

TUGAS AKHIR - IF184802

Analisis Kinerja IEEE WLAN 802.11p dan 802.11a pada Ad-hoc On-Demand Distance Vector (AODV) di lingkungan Mobile Ad-hoc Network (MANET)

Rezky Budi Prasetyo
NRP 05111340000119

Dosen Pembimbing
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - IF184802

Analisis Kinerja IEEE WLAN 802.11p dan 802.11a pada Ad-hoc On-Demand Distance Vector (AODV) di lingkungan Mobile Ad-hoc Network (MANET)

Rezky Budi Prasetyo
NRP 05111340000119

Dosen Pembimbing
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.

Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - IF184802

Analysis IEEE WLAN 802.11p and 802.11a with Ad-hoc On-Demand Distance Vector (AODV) on Mobile Ad-hoc Network (MANET)

Rezky Budi Prasetyo
NRP 05111340000119

Advisor
Dr.Eng. RADITYO ANGGORO, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Analisis Kinerja IEEE WLAN 802.11p dan 802.11a pada Ad-hoc On-Demand Distance Vector (AODV) di lingkungan Mobile Ad-hoc Network (MANET)

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur Jaringan Komputer
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

Rezky Budi Prasetyo
NRP : 05111340000119

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr.Eng. RADITYO ANGGORO, S.Kom.....
M.Sc.
NIP: 198410162008121002

(pembimbing 1)



**SURABAYA
DECEMBER 2018**

[Halaman ini sengaja dikosongkan]

Analisis Kinerja IEEE WLAN 802.11p dan 802.11a pada Ad-hoc On-Demand Distance Vector (AODV) di lingkungan Mobile Ad-hoc Network (MANET)

Nama Mahasiswa : Rezky Budi Prasetyo
NRP : 05111340000119
Jurusan : Informatika FTIK-ITS
Dosen Pembimbing 1 : Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
Dosen Pembimbing 2 : -

ABSTRAK

Perkembangan teknologi wireless dewasa ini berkembang dengan pesat. Salah satunya adalah teknologi jaringan Mobile Ad-hoc Network (MANET). Dengan jaringan MANET memungkinkan komunikasi tanpa bergantung pada ketersediaan infrastruktur yang tetap. Jaringan MANET adalah jaringan wireless yang terdiri dari kumpulan Node-node yang bergerak yang memungkinkan untuk melakukan komunikasi secara langsung.

Terdapat beberapa metode routing pada jaringan MANET diantaranya routing protocol AODV (Ad-hoc On-Demand Distance Vector). Uji coba dilakukan dengan membuat pola traffic koneksi dan pola pergerakan node yang kemudian disimulasikan dengan menggunakan script tcl protokol routing AODV.

Pada proses simulasi dijalankan menggunakan standar WLAN 802.11p dan 802.11a. Proses tersebut akan menghasilkan file output berupa trace file. Trace file hasil dari simulasi akan dianalisis untuk menghitung Packet Delivery Ratio(PDR), Routing Overhead(RO), dan End-to-End Delay.

Kata kunci: MANET, AODV, NS-2, WLAN 802.11p dan 802.11a

[Halaman ini sengaja dikosongkan]

Analysis IEEE WLAN 802.11p and 802.11a with Ad-hoc On-Demand Distance Vector (AODV) on Mobile Ad-hoc Network (MANET)

Student Name : Rezky Budi Prasetyo
Student ID : 05111340000119
Major : Informatics FTIK-ITS
Advisor 1 : Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
Advisor 2 : -

ABSTRACT

The development of wireless technology today is growing rapidly. One of them is Mobile Ad-hoc Network (MANET) network technology. The MANET network allows communication without relying on the availability of a fixed infrastructure. MANET network is a wireless network that consists of a collection of Node-moving nodes that make it possible to communicate directly.

There are several routing methods on the MANET network including AODV routing protocol (Ad-hoc On-Demand Distance Vector). The trial is done by making a pattern of connection traffic and the pattern of node movement which is then simulated by using the script tcl AODV routing protocol.

In the simulation process, the WLAN 802.11p and 802.11a standards are run. This process will produce an output file in the form of a trace file. Trace file results from the simulation will be analyzed to calculate the Packet Delivery Ratio (PDR), Overhead Routing (RO), and End-to-End Delay.

Keyword : MANET,AODV,NS-2,802.11p,802.11a

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul :

“Analisis Kinerja IEEE WLAN 802.11p dan 802.11a pada Ad-hoc On-Demand Distance Vector (AODV) di lingkungan Mobile Ad-hoc Network (MANET)”

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT atas segala nikmat dan rizki yang telah di berikan kepada penulis.
2. Keluarga yang senantiasa memberikan dukungan penuh baik secara moriil maupun materiil.
3. Bapak Dr.Eng Radityo Anggoro S.Kom., M,Sc. selaku dosen pembimbing pertama yang telah bersedia meluangkan waktu untuk membimbing dan memotivasi penulis serta memberi arahan dalam mengerjakan tugas akhir.
4. Bapak Dr.Eng. Darlis Herumurti, S.Kom, M.Sc. selaku dosen wali dari penulis yang memberikan arahan selama menjalani perkuliahan.
5. Bapak/Ibu dosen, staf dan karyawan Jurusan Teknik Informatika ITS yang telah banyak memberikan dukungan, ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
6. Teman-teman C.H.O yang selalu *men-support* penulis dalam mengerjakan tugas akhir ini

7. Teman-teman Kontrakan berprestasi yang selalu *support* penulis dalam mengerjakan tugas akhir ini.
8. Teman Teman departemen PSDM HMTTC 2014~2015 dan 2015~2016 yang telah memberikan pengalaman berarti selama kuliah.
9. Teman Teman angkatan 2013(C1D) dan angkatan 2014(C1E) yang sudah *support* penulis selama di kampus.
10. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, 11 Januari 2019
Penulis

Rezky Budi Prasetyo

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
1 BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Permasalahan	2
1.3. Batasan Permasalahan	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Metodologi.....	3
1.6.1. Penyusunan proposal Tugas Akhir	3
1.6.2. Studi literatur	4
1.6.3. Implementasi	4
1.6.4. Pengujian dan Evaluasi	4
1.6.5. Penyusunan Buku Tugas Akhir	5
1.7. Sistematika Penulisan	5
2 BAB II TINJAUAN PUSTAKA	7
2.1. Mobile Ad Hoc Network (MANET)	7
2.2. Ad Hoc On-Demand Distance Vector (AODV).....	8
2.3. IEEE WLAN 802.11p Wireless Access Vehicular Environment (WAVE)	10
2.4. IEEE WLAN 802.11a.....	10
2.5. Network Simulator 2 (NS-2).....	11
2.6. AWK.....	12
2.7. Generator File Node Movement.....	12
2.8. File Traffic Connection Pattern	14
3 BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15

3.1.	Deskripsi Umum Sistem	15
3.2.	Perancangan Skenario.....	17
3.2.1.	Perancangan Skenario Node Movement (Mobility Generation).....	17
3.2.2.	<i>Traffic-Connection Pattern</i>	18
3.3.	Perancangan Simulasi NS-2.....	19
3.4.	Perancangan Metriks Analisis	19
3.4.1.	<i>Packet Delivery Ratio</i> (PDR)	19
3.4.2.	<i>End-to-End Delay</i> (E2E).....	20
3.4.3.	<i>Routing Overhead</i> (RO).....	20
4	BAB IV IMPLEMENTASI.....	21
4.1.	Lingkungan Implementasi Protokol	21
4.2.	Implementasi Skenario	21
4.2.1.	Skenario <i>File Node-Movement</i> (Mobility Generation)	22
4.2.2.	<i>File traffic-connection pattern</i>	23
4.3.	Implementasi Simulasi pada NS-2	24
4.4.	Implementasi Metriks Analisis.....	30
4.4.1.	Implementasi <i>Packet Delivery Ratio</i>	30
4.4.2.	Implementasi <i>End-to-End Delay</i>	31
4.4.3.	Implementasi <i>Routing Overhead</i>	32
5	BAB V PENGUJIAN DAN EVALUASI	33
5.1.	Lingkungan Pengujian.....	33
5.2.	Kriteria Pengujian.....	33
5.3.	Analisis Packet Delivery Ratio (PDR).....	34
5.4.	Analisis Routing Overhead(RO)	37
5.5.	Analisis End-to-End Delay (E2E)	40
6	BAB VI KESIMPULAN DAN SARAN	45
6.1.	Kesimpulan	45
6.2.	Saran.....	46
7	DAFTAR PUSTAKA	49
	LAMPIRAN	51
	BIODATA PENULIS	67

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi skenario MANET [2].....	7
Gambar 3.1 Skema umum system	16
Gambar 4.1 Contoh hasil nilai PDR	31
Gambar 4.2 Contoh hasil nilai E2E	31
Gambar 4.3 Contoh hasil nilai RO.....	32
Gambar 5.1 Grafik nilai PDR 802.11p dan 802.11a dengan node static50	35
Gambar 5.2 Grafik nilai PDR 802.11p dan 802.11a dengan speed static10	36
Gambar 5.3 Grafik nilai RO 802.11p dan 802.11a dengan node statik50.....	38
Gambar 5.4 Grafik nilai RO 802.11p dan 802.11a dengan speed statik10.....	39
Gambar 5.5 Grafik hasil nilai E2E dengan node statik50	41
Gambar 5.6 Grafik hasil nilai E2E dengan speed statik10	42

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Penjelasan <i>Command Line</i> ‘setdest’	13
Tabel 2.2 Penjelasan <i>Command Line</i> ‘cbrgen.tcl’	14
Tabel 3.1 Tabel Istilah	17
Tabel 3.2 Penjelasan Skenario <i>node movement</i>	18
Tabel 3.3 Penjelasan <i>Traffic-connection pattern</i>	18
Tabel 3.4 Penjelasan simulasi NS-2	19
Tabel 4.1 Penjelasan simulasi NS-2	24
Tabel 5.1 Penjelasan skenario pengujian	33
Tabel 5.2 Nilai hasil PDR 802.11p node50 statik	34
Tabel 5.3 Nilai hasil PDR 802.11a node50 statik	34
Tabel 5.4 Nilai hasil PDR 802.11p speed10 statik	36
Tabel 5.5 Nilai hasil PDR 802.11a speed10 statik	36
Tabel 5.6 Nilai hasil RO 802.11p node50 statik	37
Tabel 5.7 Nilai hasil RO 802.11a node50 statik	38
Tabel 5.8 Nilai hasil RO 802.11p speed10 statik	39
Tabel 5.9 Nilai hasil RO 802.11a node50 statik	39
Tabel 5.10 Hasil nilai E2E 802.11p node50 statik	40
Tabel 5.11 Hasil nilai E2E 802.11a node50 statik	40
Tabel 5.12 Hasil nilai E2E 802.11p speed10 statik	42
Tabel 5.13 Hasil nilai E2E 802.11a speed10 statik	42

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 2.1 Format <i>Command Line</i> ‘setdest’	12
Kode Sumber 2.2 Contoh <i>Command Line</i> ‘setdest’	13
Kode Sumber 2.3 Format <i>Command Line</i> ‘cbrgen.tcl’	14
Kode Sumber 2.4 Contoh <i>Command Line</i> ‘cbrgen.tcl’	14
Kode Sumber 4.1 Format kode ‘setdest’	22
Kode Sumber 4.2 Implementasi ‘setdest’	23
Kode Sumber 4.3 Format kode ‘cbrgen.tcl’	23
Kode Sumber 4.4 Implementasi kode ‘cbrgen.tcl’	24
Kode Sumber 4.5 Konfigurasi parameter pada NS-2.....	25
Kode Sumber 4.6 Pengaturan <i>Transmission range</i>	25
Kode Sumber 4.7 Pengaturan variabel global pada NS-2	26
Kode Sumber 4.8 Pengaturan IEEE 802.11p	28
Kode Sumber 4.9 Pengaturan IEEE 802.11a.....	29
Kode Sumber 4.10 Menginisiasi penempatan awal node.....	30
Kode Sumber 4.11 Menjalankan file PDR.awk.....	31
Kode Sumber 4.12 Menjalankan file E2E.awk	31
Kode Sumber 4.13 Menjalankan file ro.awk.....	32
Kode Sumber 7.1 Posisi <i>node</i> dari potongan Skenario	53
Kode Sumber 7.2 Pembuatan ‘GOD’ dari potongan skenario	55
Kode Sumber 7.3 Koneksi yang digunakan pada ‘cbr.txt’	56
Kode Sumber 7.4 file .tcl untuk simulasi AODV 802.11p.....	60
Kode Sumber 7.5 file .tcl untuk simulasi AODV 802.11a	63
Kode Sumber 7.6 Implementasi perhitungan RO	64
Kode Sumber 7.7 implementasi perhitungan PDR	65
Kode Sumber 7.8 implementasi perhitungan E2E.....	66
Kode Sumber 7.9 instalasi NS-2.....	66
Kode Sumber 7.10 mengunduh kode sumber ns-2.35	66
Kode Sumber 7.11 Ekstrak file NS-2.....	66

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Perkembangan teknologi informasi dan komunikasi sudah berkembang dengan sangat pesat, jauh melewati ekspektasi orang-orang. Perkembangan ini ditunjukkan dengan terciptanya berbagai macam teknologi yang membantu meningkatkan segi produktivitas manusia dalam kehidupan sehari-harinya.

Salah satu teknologi yang memudahkan manusia untuk saling berkomunikasi adalah Mobile Ad-hoc Network (MANET). MANET memungkinkan terjadinya komunikasi jaringan tanpa bergantung pada ketersediaan infrastruktur jaringan yang tetap. Jaringan MANET merupakan jaringan *wireless* yang terdiri dari kumpulan *Node-node* yang bergerak yang memungkinkan untuk melakukan komunikasi secara langsung.

Dalam teknologi MANET tidak bisa dilepaskan dengan proses pengiriman paket data. Proses pencarian rute untuk mengirimkan data dari sumber ke tujuan disebut dengan *routing*. Dalam proses *routing*, data yang berasal dari *Node* sumber dikirimkan ke *Node-node* lain hingga mencapai node tujuan. Terdapat beberapa metode *routing* pada jaringan MANET salah satunya *Ad-hoc On-Demand Distance Vector* (AODV). Protokol *routing* AODV bersifat reaktif yang artinya hanya melakukan proses pembentukan rute pada saat dibutuhkan. Setiap kali pencarian rute, AODV hanya menemukan satu pilihan rute atau *multipath*. Apabila terjadi kegagalan rute, maka AODV akan mengulang proses pencarian rute. Untuk mengatasi hal tersebut, dibutuhkan protokol *routing* yang memiliki beberapa pilihan rute atau *multipath* dalam setiap pencarian rute.

Implementasi pada lingkungan MANET dapat dilakukan dengan menggunakan simulasi sehingga penelitian ini dapat dilakukan untuk mempelajari sistem dengan baik. Simulasi dilakukan dengan menggunakan *Network Simulator 2* (NS-2). Implementasi ini akan dilakukan analisa kinerja dari IEEE 802.11p dan 802.11a pada protokol *routing* AODV berdasarkan *Packet Delivery Ratio*(PDR), *Routing Overhead*(RO), dan *End-to-End Delay*.

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana kinerja Protokol *routing* AODV dengan 802.11 pada Manet?
2. Bagaimana hasil analisis studi kinerja IEEE 802.11p dan 802.11a pada Protokol *routing* AODV di lingkungan MANET?

1.3. Batasan Permasalahan

Beberapa batasan masalah yang terdapat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Protokol *Routing* yang diuji coba adalah *Ad-hoc On-Deman Distance Vector* (AODV).
2. *MAC protocol* yang akan dianalisis dalam Tugas Akhir ini adalah IEEE 802.11p dan 802.11a.
3. Skenario Ujicoba dilakukan pada Topologi MANET.
4. Simulasi Pengujian Protocol *Routing* Menggunakan *Network Simulator 2*.
5. Analisis Kinerja Didasarkan pada *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay*.

1.4. Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah menganalisis IEEE 802.11p dan 802.11a pada *protocol* AODV yang efisien dengan parameter *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay*.

1.5. Manfaat

Manfaat Tugas Akhir ini diharapkan dapat menghasilkan hasil analisis kinerja IEEE 802.11p dan 802.11a pada Protokol AODV yang efisien dengan parameter *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), *End-to-End Delay*.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1.6.1. Penyusunan proposal Tugas Akhir

Proposal tugas akhir ini berisi mengenai rencana Melakukan analisa pada IEEE 802.11p dan 802.11a pada *protocol* AODV di lingkungan MANET. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula subbab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.6.2. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan Tugas Akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik Tugas Akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. MANET
2. NS-2
3. AODV
4. IEEE 802.11p
5. IEEE 802.11a

1.6.3. Implementasi

Pada tahap ini dilakukan implementasi berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini merupakan yang paling penting dimana bentuk awal aplikasi yang diimplementasikan didefinisikan. Pada tahapan ini dibuat prototype system yang merupakan rancangan dasar dari system yang akan dibuat. Serta dilakukan desain suatu system dan desain proses proses yang ada

1.6.4. Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap system yang sudah dibuat. Pengujian dan evaluasi akan diketahui berdasarkan nilai dari *Paket Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *End-to-End Delay*.

1.6.5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7. Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Tinjauan Pustaka

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan metode yang nantinya akan diimplementasikan dan dilakukan pengujian dari aplikasi yang dibangun.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan sistem yang sudah dilakukan pada tahap perancangan. Penjelasan berupa skenario *node node* pada jaringan *wireless*, konfigurasi sistem dan skrip analisis yang digunakan untuk menguji performa protocol *routing*

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari sistem dan performa dalam skenario mobilitas *ad hoc* yang dibuat dalam

network simulator.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan tentang tinjauan pustaka yang menjadi dasar pembuatan Tugas Akhir. Beberapa teori, pustaka, dan teknologi yang mendasari pengerjaan Tugas Akhir ini. Serta memberi gambaran terhadap alat, protokol *routing* serta definisi yang digunakan dalam pembuatan Tugas Akhir.

2.1. *Mobile Ad Hoc Network* (MANET)

Mobile Ad-Hoc Network (MANET) merupakan kumpulan dari perangkat bergerak atau *mobile device* yang dapat digunakan menjadi sekumpulan *node* tanpa infrastuktur yang terpusat. Sifat dinamis yang dimiliki MANET sendiri membuat setiap *node* yang ada di dalam jaringan tersebut berperilaku sebagai *router* yang mana memiliki peran dalam penemuan dan pemeliharaan rute pengiriman paket ke *node* lain dalam jaringan. Keuntungan utama dalam penggunaan MANET sendiri adalah bersifat fleksibel, biaya yang rendah, dan bersifat tangguh. Keunggulan MANET ini memiliki implementasi yang penting terhadap operasi militer di wilayah perang, operasi penyelamatan sandera perang dan menyebarkan informasi secara cepat dan menyeluruh ketika perang[1].

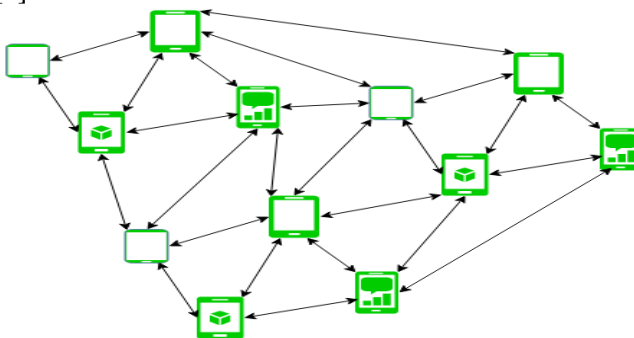


Figure - Mobile Ad Hoc Network

Gambar 2.1 Ilustrasi skenario MANET [2]

Protokol *routing* yang tersedia dalam MANET terbagi menjadi tiga kategori yaitu.

- *Proactive routing*: jenis *routing* ini bersifat menyebarkan seluruh informasi rute tabel ke setiap *node* yang ada. Apabila terjadi perubahan maka informasi perubahannya akan disebarkan ke seluruh *node*. Contoh dari tipe *routing* ini adalah *Optimized Link State Routing* (OLSR).
- *Reactive routing*: cara kerja dari *routing* ini adalah apabila *node* sumber mau mengirimkan paket maka *node* sumber akan mengirimkan *route request*. setelah sampai di *node* tujuan maka akan di kirim *node reply* ke *node* sumber. Contoh dari *routing* ini adalah *Dynamic Source Routing* (DSR), *ad-hoc on Demand Vector*(AODV).
- *Hybrid Routing*: jenis *routing* ini merupakan gabungan dari *Proactive routing* dan *Reactive routing*. Contoh dari *routing* ini adalah *Zone Routing Protocol*(ZRP).

2.2. *Ad Hoc On-Demand Distance Vector (AODV)*

Ad Hoc On-Demand Distance Vector merupakan sebuah *routing protocol* yang bersifat reaktif, hal ini berarti protocol akan mengolah dan memproses rute hanya ketika melakukan request saja. Berdasarkan RFC 3561, terdapat dua proses utama dalam AODV, yaitu *route discovery* dan *route maintenance*. *Route discovery* dilakukan dengan menginisiasi paket *route request* (RREQ) dan diikuti sebuah *route reply* (RREP) yang akan diberikan oleh *node* tujuan ketika sudah menerima paket RREQ. Selain paket RREQ dan RREP, terdapat paket *route error* (RERR) yang membantu dalam proses *route maintenance*[3].

Pada setiap *node* yang menggunakan *routing protocol* AODV pasti memiliki sebuah *routing Tabel* dengan *field* sebagai berikut:

- *Destination Address*: berisi alamat dari *node* tujuan.
- *Destination Sequence Number*: *sequence number* dari jalur komunikasi.
- *Next Hop*: alamat *node* yang akan meneruskan paket data.
- *Hop Count*: jumlah *hop* yang harus dilakukan agar paket dapat mencapai *node* tujuan.
- *Lifetime*: waktu dalam milidetik yang diperlukan node untuk menerima RREP.
- *Routing Flags*: status jalur. Terdapat tiga tipe status, yaitu up (valid), down (tidak valid) atau sedang diperbaiki.

2.3. IEEE WLAN 802.11p Wireless Access Vehicular Environment (WAVE)

Untuk memanfaatkan potensi-potensi yang ada pada sistem komunikasi, saat ini IEEE sedang mengembangkan suatu perubahan standar IEEE 802.11p atau biasa disebut dengan WAVE (Wireless Access Vehicular Environment). WAVE merupakan penyempurnaan standar IEEE 802.11 yang diperlukan untuk mendukung pengaplikasian ITS (Intelligent Transportation Systems).

WAVE juga merupakan pengembangan sistem IEEE 802.11a dengan memperkenalkan *physical layer* dan *MAC layer* yang dapat meningkatkan sistem operasi dan aplikasi keselamatan dengan memberikan tingkat *latency* rendah. WAVE sendiri beroperasi pada *bandwidth* 5.9 GHz dengan menggunakan sistem *multiplexing* OFDM (Orthogonal Frequency Division Multiplexing) dan dapat mencapai kecepatan transmisi data antara 6-27Mbps

WAVE terdiri dari tujuh channel pada frekuensi 10 MHz yang dari satu *control channel* dan enam *service channel* pada *bandwidth* 5.9 GHz. *Service channel* digunakan untuk *public safety* dan *private service*, sedangkan *control channel* digunakan sebagai referensi channel untuk membangun link komunikasi antara RSU (Road-Side Unit) dan OBU (On-Broad Unit) [4].

2.4. IEEE WLAN 802.11a

sebuah teknologi jaringan nirkabel yang merupakan pengembangan lebih lanjut dari standar IEEE 802.11 yang asli, namun bekerja pada *bandwidth* 5.8 GHz dengan kecepatan maksimum hingga 54 Mb/s. Metode transmisi yang digunakan adalah *Orthogonal Frequency Division Multiplexing* (OFDM), yang memungkinkan transmisi data secara paralel di dalam sub-frekuensi. Penggunaan OFDM memiliki keunggulan resistansi terhadap interferensi dengan gelombang lain, dan tentunya

peningkatan throughput. Standar ini selesai diratifikasi pada tahun 1999[5].

2.5. *Network Simulator 2 (NS-2)*

NS2 adalah alat simulasi jaringan *open source* yang banyak digunakan dalam mempelajari struktur dinamik dari jaringan komunikasi. Simulasi dari jaringan nirkabel dan protokol (seperti algoritma routing, TCP, dan UDP) dapat diselesaikan dengan baik dengan simulator ini. Karena kefleksibelannya, NS2 menjadi populer dikalangan komunitas peneliti sejak awal kemunculannya pada tahun 1989[6].

NS2 terdiri dari dua bahasa pemrograman yaitu C++ dan OTcl(*Objek-oriented Tool Command Language*). C++ mendefinisikan mekanisme internal dari simulasi objek dan OTcl berfungsi untuk menset simulasi dengan assembly dan mengkonfigurasi objek sebagai penjadwalan diskrit. C++ dan OTcl saling berhubungan menggunakan TclCL. Setelah simulasi, output NS2 dapat berupa basis teks atau animasi berdasarkan simulasi. Untuk menginterpretasikan output ini secara grafik dan interaktif maka dibutuhkan NAM (Network Animator) dan XGraph. Untuk menganalisa tingkah laku dari jaringan user dapat mengekstrak subset dari data teks dan mentransformasikannya agar menjadi lebih atraktif.

Pada prakteknya, NS2 merupakan simulasi yang berjalan pada sistem UNIX. Oleh sebab itu NS2 dapat berjalan dengan baik di sistem operasi Linux, OpenBSD, FreeBSD, dan sistem operasi berbasis unix lainnya. Walaupun demikian NS2 dapat juga berjalan pada Windows dengan menggunakan tool tambahan yaitu Cygwin. Cygwin adalah *port* dari *tool* pengembangan GNU (*GNU's Not UNIX*) untuk Microsoft Windows

2.6. AWK

AWK merupakan sebuah pemrograman seperti pada shell atau C yang memiliki karakteristik yaitu sebagai alat yang cocok untuk memanipulasi sebuah text yang dapat digunakan sebagai ekstraksi dari sebuah *dataset*. AWK ditulis menggunakan Bahasa pemrogramannya sendiri yaitu *awk programming language*.

Pada tugas akhir ini penulis menggunakan AWK sebagai alat untuk membuat script dalam perhitungan *Packet Delivery Ratio* (PDR), *End-to-End Delay* (E2D), dan *Routing Overhead* (RO) dari hasil *trace* NS-2[7].

2.7. Generator File Node Movement

CMU atau *Carnegie Mellon University* mengembangkan sebuah alat bernama ‘setdest’ yang digunakan untuk men-*generate random movement* dari *node* di jaringan *wireless*. *Node movement* dihasilkan dengan kecepatan maksimal yang spesifik serta penempatan setiap *node* nya yang acak ataupun yang telah ditentukan. Apabila *node* yang sudah ditentukan sampai pada lokasi tujuan maka *node* tersebut akan berpindah ke lokasi selanjutnya. Pergerakan *node* tersebut dapat dihentikan sementara.

Sebelum menjalankan program simulasi pengguna harus menjalankan program ‘setdest’. Format *command line* ‘setdest’ ditunjukkan pada Kode Sumber 2.1 dan keterangannya ditunjukkan pada tabel 2.1.

```
1. ./setdest [-v version] [-n num_of_nodes] [-p pausetime] [-M maxspeed] [-t simtime] [-x maxx] [-y maxy] > [outdir/movement-file]
```

Kode Sumber 2.1 Format Command Line ‘setdest’

Tabel 2.1 Penjelasan *Command Line* ‘setdest’

Parameter	Keterangan
-v <i>version</i>	Menentukan versi dari ‘Setdest’ yang digunakan
-n <i>num</i>	Menentukan jumlah node yang dibuat pada skenario
-p <i>pausetime</i>	Menentukan durasi berhenti pada sebuah paket apabila sudah sampai di tujuan
-M <i>maxspeed</i>	Menentukan kecepatan maksimum dari pengiriman paket.
-t <i>simtime</i>	Menentukan lama waktu jalannya simulasi.
-x <i>max x</i>	Menentukan panjang maksimum dari area simulasi.
-y <i>max y</i>	Menentukan lebar maksimum dari area simulasi

Command line ‘setdest’ yang sudah di-generate menghasilkan *file* yang berisi jumlah *node* dan pergerakan dari *node* yang sudah dibuat dalam *file* berbentuk .tcl. Selain pergerakan *node file* tersebut juga berisi tentang perpindahan rute[8].

Untuk membuat skenario *node-movement* yang terdiri dari 50 *node*, bergerak dengan kecepatan maksimum 5 m/s, simulasi akan berhenti setelah 100 detik dengan batas topologi yang diartikan sebagai 800 x 800 meter, contoh *command line* seperti pada Kode Sumber 2.2.

1. `./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x 800 -y 800 > scenario.txt`

Kode Sumber 2.2 Contoh *Command Line* ‘setdest’

2.8. File Traffic Connection Pattern

Random Traffic Connection memiliki dua buah tipe *traffic* yaitu TCP dan CBR. Keduanya dapat dibuat *menggunakan script traffic scenario generator*. *Script* ini dapat membantu kita untuk *men-generate* beban *traffic*. *Script* yang dimaksud di dalam sini adalah ‘cbrgen.tcl’. program “cbrgen.tcl” diiugunakan sesuai dengan *command line* pada gambar 2.4 dan dengan keterangan yang diunjukkan pada tabel 2.2.[9]

```
1. ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-
    seed seed] [-mc connections] [-
    rate rate] > traffic-
```

Kode Sumber 2.3 Format Command Line ‘cbrgen.tcl’

Tabel 2.2 Penjelasan Command Line ‘cbrgen.tcl’

Parameter	Keterangan
-type cbr tcp	Menentukan jenis <i>traffic</i> yang digunakan.
-nn <i>nodes</i>	Menentukan jumlah total <i>node</i> .
-s <i>seed</i>	Menentukan nilai <i>random seed</i>
-mc <i>connection</i>	Menentukan jumlah koneksi antar node.
-rate <i>rate</i>	Menentukan jumlah paket per detik yang terkirim.

```
1. ns cbrgen.tcl -type cbr -nn 2 -seed 1.0 -mc 1 -
    rate 0.25 > cbr.txt
```

Kode Sumber 2.4 Contoh Command Line ‘cbrgen.tcl’

BAB III

ANALISIS DAN PERANCANGAN SISTEM

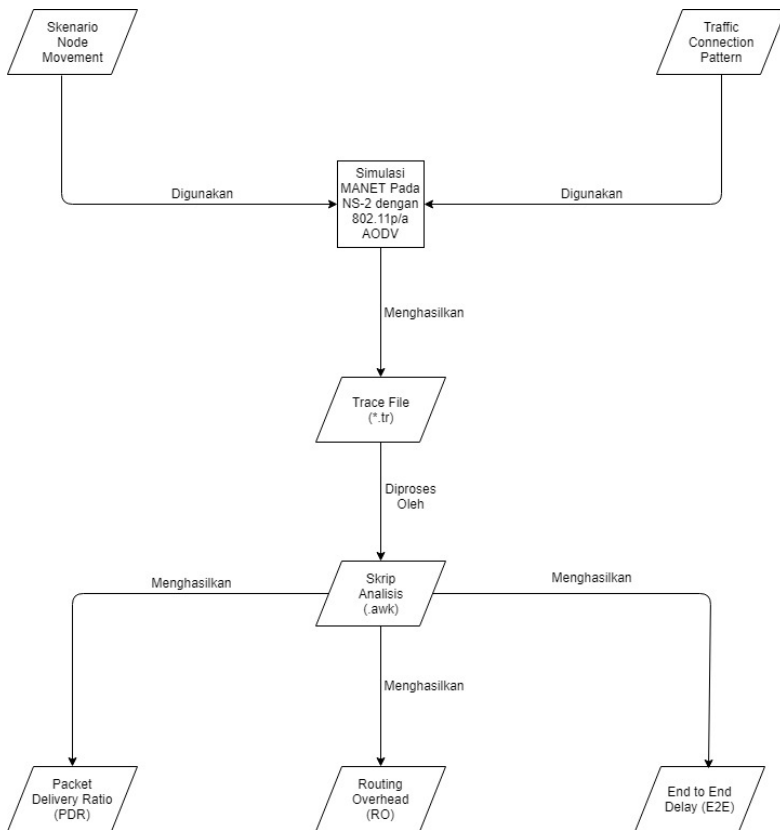
Pada bab ini akan dijelaskan mengenai dasar perancangan dari perangkat lunak yang akan dibangun dalam tugas akhir ini. Perancangan tersebut mencakup deskripsi umum aplikasi, arsitektur sistem, model fungsional dan diagram alur aplikasi.

3.1. Deskripsi Umum Sistem

Pada Tugas Akhir ini akan dilakukan analisis mengenai performa IEEE 802.11p dan 802.11a pada *routing protocol* AODV di lingkungan MANET. pada pembuatan skenario penulis menggunakan *mobility generator* yang bersifat random way point dan telah ada pada *Network Simulator-2* (NS-2) yaitu dengan cara *men-generate file node movement* dan membuat koneksi antar *node* menggunakan *file traffic connection pattern*. Rancangan Simulasi yang dibuat dapat dilihat pada Gambar 3.1.

Dalam penelitian ini penulis akan menganalisa performa dari IEEE 802.11p dan 802.11a pada simulasi skenario yang dijalankan pada NS-2 menggunakan *routing protocol* AODV pada system operasi Linux.

Pada tiap skenario, kecepatan maksimum dari satu *node* ke *node* lainnya dibuat bervariasi yaitu 5, 10 dan 15 m/s begitu dengan jumlah node dibuat bervariasi yaitu 50, 75, dan 100. Hasil proses uji coba dari tiap skenario akan menghasilkan sebuah *trace file* yang nantinya akan dilakukan analisis perhitungan metrik *packet delivery Ratio* (PDR), *End-to-End Delay* (E2E), dan *Routing Overhead* (RO). Dari hasil metrik tersebut dianalisis performa IEEE 802.11p dan 802.11a.



Gambar 3.1 Skema umum system

Tabel 3.1 Tabel Istilah

Istilah	Keterangan
MANET	<i>Mobile Ad-Hoc Network</i>
AODV	<i>Ad Hoc On-Demand Distance Vector</i>
PDR	<i>Packet Delivery Ratio</i>
RO	<i>Routing OverHead</i>
E2E	<i>End-to-End Delay</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
CBR	<i>Constant bitrate</i>

3.2. Perancangan Skenario

Skenario uji coba dalam Tugas Akhir ini dibuat dengan menggunakan *mobility generation* setelah itu penguji akan membuat koneksi dari skenario yang sudah ada dengan *file traffic connection* yang sudah ada pada NS-2. Pada tugas akhir ini skenario dibuat untuk melihat pergerakan dari *node* yang sudah dibuat berdasarkan pada tiga jumlah *node* yaitu 50, 75 dan 100 dan tiga kecepatan maksimal yaitu 5m/s, 10m/s, 15m/s. Sedangkan untuk koneksinya digunakan dua *node* untuk menentukan *node* pengirim dan *node* penerima paket.

3.2.1. Perancangan Skenario Node Movement (Mobility Generation)

Perancangan skenario dibuat dengan men-*generate file node movement* yang sudah disediakan oleh NS-2 biasa kita kenal dengan *tools* bernama 'setdest' yang akan digunakan dalam *file tcl* selama simulasi pada NS-2 sebagai bentuk pergerakan *node* yang berpindah-pindah.

Tabel 3.2 Penjelasan Skenario *node movement*

No.	Parameter	Spesifikasi
1	Jumlah <i>Node</i>	-50 -75 -100
2	Waktu Simulasi	100 detik
3	Area	800m x 800m
4	Kecepatan Maksimal	-5m/s -10m/s -15m/s
5	Sumber <i>Traffic</i>	CBR
6	Waktu Jeda (Dalam Detik)	1
7	Ukuran Paket	512 bytes
8	<i>Rate</i> Paket	0.25 paket per detik
9	Jumlah maksimal koneksi	1
10	Model mobilitas yang digunakan	<i>Random way point</i>

3.2.2. *Traffic-Connection Pattern*

Traffic-Connection dibuat dengan menjalankan program *cbrgren.tcl* yang telah ada pada NS-2 yang digunakan untuk memberikan koneksi dari *node node* yang sudah dibuat dalam skenario diatas selama melakukan simulasi pada NS-2.

Tabel 3.3 Penjelasan *Traffic-connection pattern*

No.	Parameter	Spesifikasi
1	-type cbr tcp	CBR
2	-nn <i>nodes</i>	2
3	-s <i>seed</i>	1.0
4	-mc <i>connection</i>	1
5	-rate <i>rate</i>	0.25

3.3. Perancangan Simulasi NS-2

Pada perancangan kode NS-2 dengan konfigurasi MANET. Dilakukan dengan menggunakan skenario mobilitas dan digabungkan dengan *script* TCL yang berisikan konfigurasi mengenai lingkungan simulasi. Konfigurasi lingkungan simulasi MANET pada NS-2 dapat dilihat di tabel 3.2

Tabel 3.4 Penjelasan simulasi NS-2

No	Parameter	Spesifikasi
1	<i>Network Simulator</i>	NS-2, 2.35
2	<i>Routing Protocol</i>	AODV
3	Waktu Simulasi	100 detik
4	Area Simulasi	800m x 800m
5	Banyak <i>node</i>	-50 -75 -100
6	Radius Transmisi	250m
7	Agen Pengirim	<i>Constant Bit Rate</i> (CBR)
8	<i>Source/Destination</i>	Statis
9	<i>Packet Rate</i>	512 bytes
10	Ukuran Paket	32
11	Protokol MAC	IEEE 802.11p/a
12	Propagasi Sinyal	<i>TwoRayGround</i>
13	Tipe Kanal	<i>Wireless Channel</i>

3.4. Perancangan Metriks Analisis

Berikut ini merupakan beberapa parameter yang dianalisis dalam Tugas Akhir ini:

3.4.1. *Packet Delivery Ratio* (PDR)

Packet delivery ratio merupakan perbandingan dari jumlah paket komunikasi yang dikirimkan dengan paket komunikasi yang diterima. *Packet delivery ratio* dihitung menggunakan persamaan

2, dimana *received* adalah jumlah paket komunikasi yang diterima dan *sent* adalah jumlah paket komunikasi yang dikirimkan. Semakin tinggi *packet delivery ratio* semakin berhasil pengiriman paket yang dilakukan.

$$Packet\ Delivery\ Ratio = \frac{Received}{Sent} \quad (1)$$

3.4.2. End-to-End Delay(E2E)

End-to-end delay mengindikasikan interupsi transmisi paket dari *node* asal ke tujuan. Total interupsi didapatkan dari akumulasi *delay-delay* kecil yang ada dalam jaringan. Total interupsi terdiri dari *delay* yang mungkin karena *buffer* pada *route discovery latency*, *delay* pada antrian *interface*, retransmisi. Rata-rata *end to end delay* pada paket yang diterima bisa dihitung berdasarkan selisih waktu antara transmisi dan respon paket pada *Constant Bit Rate* (CBR) dan membaginya dengan jumlah total transimi CBR menggunakan persamaan 3

$$End\ to\ End\ Delay = \sum_{i < sent}^{i=0} \frac{t^{received(i)} - t^{sent(i)}}{sent} \quad (2)$$

3.4.3. Routing Overhead (RO)

Routing overhead adalah jumlah paket *routing* yang ada didalam sebuah jaringan dibagi dengan jumlah keseluruhan paket data yang diterima, perhitungan menggunakan persamaan 4.

$$Routing\ Overhead = \sum_{i < sent}^{i=0} \frac{packet\ routing}{received} \quad (3)$$

BAB IV IMPLEMENTASI

Bab ini akan menjelaskan tentang implementasi Tugas Akhir berdasarkan rancangan perangkat lunak yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi skenario, implementasi simulasi NS-2, dan implementasi matiks analisis.

4.1. Lingkungan Implementasi Protokol

Sub bab ini menjelaskan tentang lingkungan implementasi Protokol yang dilakukan pada lingkungan:

Tabel 4.1 Spesifikasi Lingkungan Implementasi

Perangkat Keras	- Laptop Asus ROG GL502V <ul style="list-style-type: none">o Prosesor Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHzo RAM 16 GBo HDD 1 TB
Perangkat Lunak	- Laptop Asus ROG GL502V <ul style="list-style-type: none">o Sistem Operasi Ubuntu 14.04o Network Simulator 2, 2.35

4.2. Implementasi Skenario

Implementasi skenario mobilitas MANET dimulai dengan melakukan pembuatan skenario oleh *Mobility Generation* untuk menempatkan dan menentukan *node node* yang akan di simulasikan. Setelah membuat skenario maka kita membuat jalur jalur yang menghubungkan antar node dengan *traffic connection pattern*. implementasi skenarionya adalah sebagai berikut.

4.2.1. Skenario *File Node-Movement(Mobility Generation)*

Dalam implementasi yang dilakukan pada pembuatan skenario dengan *mobility generation* menggunakan *tools generate default* yang sudah disediakan oleh NS-2 ketika kita melakukan installasi sebelumnya yaitu 'setdest'. Skenario yang sudah di-generate ini akan digunakan untuk setiap simulasi yang dilakukan berdasarkan kecepatan maksimal yang berbeda beda. Algoritma yang digunakan untuk membuat skenario ini adalah *Random Way Point* sehingga penempatan *node node* yang ada akan bersifat acak. *Format command line* pada kode sumber 4.1 yang digunakan untuk menghasilkan gerakan acak pada node adalah sebagai berikut:

```
1. ./setdest [-v version] [-n num_of_nodes] [-
  p pausetime] [-M maxspeed] [-t simtime] [-
  x maxx] [-y maxy] > [outdir/movement-file]
```

Kode Sumber 4.1 Format kode 'setdest'

Ketentuan-ketentuan yang diujicobakan pada skenario ini adalah versi 'setdest' simulator yaitu 1, jumlah node dalam skenario 1 yaitu 50, skenario 2 yaitu 75 dan skenario 3 yaitu 100, waktu jeda (*pause time*) yaitu 10 detik, kecepatan maksimalnya yaitu skenario A sebesar 5m/s, skenario B sebesar 10m/s dan skenario C sebesar 15m/s, waktu simulasi yaitu 100 detik. Kemudian file mobilitas yang dihasilkan disimpan dalam direktori "*~ns/indep-utils/cmu-scen-gen/setdest/*". Pada kode sumber 4.2 dapat dilihat implementasi *command line* pada 'setdest' dengan berbagai kecepatan maksimal yang berbeda sesuai dan node sebanyak 50. Dan untuk setiap kecepatan maksimal tersebut dibuat 10 buah file untuk satu protokol routing dan satu model propagasi.

```

1. ./setdest -v 1 -n 50 -p 10 -M 5 -t 100 -x 800 -
   y 800 > scenario.txt
2. ./setdest -v 1 -n 50 -p 10 -M 10 -t 100 -x 800 -
   y 800 > scenario.txt
3. ./setdest -v 1 -n 50 -p 10 -M 15 -t 100 -x 800 -
   y 800 > scenario.txt
4. ./setdest -v 1 -n 50 -p 10 -M 10 -t 100 -x 800 -
   y 800 > scenario.txt
5. ./setdest -v 1 -n 75 -p 10 -M 10 -t 100 -x 800 -
   y 800 > scenario.txt
6. ./setdest -v 1 -n 100 -p 10 -M 10 -t 100 -x 800 -
   y 800 > scenario.txt
7.

```

Kode Sumber 4.2 Implementasi ‘setdest’

4.2.2. File *traffic-connection pattern*

Dalam implementasi *random traffic connection* yang menggunakan tipe CBR ini di-setting dengan menggunakan skrip *traffic scenario generator*. skrip *traffic scenario generator* akan menghasilkan dule bernama ‘cbrgen.tcl’ yang mana akan digunakan unruk membuat jaringan atau hubungan antar *node node* yang sudah dibuat pada skenario sebelumnya. ketika kira akan menggunakan file ‘cbrgen.tcl’ ini kita harus menentukan tipe koneksi yang digunakan(apakah CBR atau TCP), banyaknya *node* yang ada, koneksi maksimal yang dimau, dan nilai *random seed*. *Format command line* pada kode sumber 4.7 yang digunakan untuk menghasilkan gerakan acak pada *node* adalah sebagai berikut.

```

1. ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-
   seed seed] [-mc connections] [-
   rate rate] > traffic-

```

Kode Sumber 4.3 Format kode ‘cbrgen.tcl’

Pada kode sumber 4.4 merupakan bentuk implementasi untuk menjalankan `cbrgen.tcl` untuk membuat file koneksi CBR diantara 2 node memiliki maksimal 1 koneksi dengan nilai seed 1.0 dan jumlah paket per detik sebanyak 0.25 yang disimpan dalam `cbr.txt` yang nantinya akan digunakan pada saat simulasi NS-2.

```
1. ns cbrgen.tcl -type cbr -nn 2 -seed 1.0 -mc 1 -
   rate 0.25 > cbr.txt
```

Kode Sumber 4.4 Implementasi kode ‘cbrgen.tcl’

4.3. Implementasi Simulasi pada NS-2

Implementasi simulasi NS-2 dilakukan dengan cara pendeskripsian lingkungan simulasi pada sebuah file dengan *ekstensi* `.tcl`, dan `.txt` file-file ini berisi konfigurasi setiap *node* dan proses yang dilakukan selama simulasi berjalan.

Pada kode sumber 4.5 menunjukkan skrip konfigurasi awal parameter parameter yang diberikan untuk menjalankan simulasi MANET pada NS-2. Isi dari parameternya adalah sebagai berikut.

Tabel 4.1 Penjelasan simulasi NS-2

No.	Parameter	Spesifikasi
1	Channel/	Wireless Channel
2	Propagation/	TwoRayGround
3	Phy/ WirelessPhy	WirelessPhyExt
4	Mac/ 802.11	802.11Ext
5	Queue Drotail/PriQueue	PriQueue
6	Antenna/OmniAntena	OmniAntena
7	Nilai X	800
8	Nilai Y	800
9	Nilai seed	0.0
10	Routing Protocol	AODV
11	File traffic connection	“cbrtest.txt”
12	File node movement	“scenario.txt”

```

1. set val(chan)           Channel/WirelessChannel;
2. set val(prop)           Propagation/TwoRayGround;
3. set val(netif)          Phy/WirelessPhyExt;
4. set val(mac)            Mac/802_11Ext;
5. set val(ifq)            CMUPriQueue;
6. set val(ll)            LL;
7. set val(ant)            Antenna/OmniAntenna;
8. set val(ifqlen)         50;
9. set val(nn)             50;
10. set val(rp)            AODV;
11. set opt(x)             800;
12. set opt(y)             800;
13. set val(stop)          100;
14. set val(seed)          0;
15. set val(cp)            "cbr.txt";
16. set val(sc)            "scenario.txt";

```

Kode Sumber 4.5 Konfigurasi parameter pada NS-2

Pada kode sumber 4.6 merupakan pengaturan dari *transmission range* yang digunakan pada simulasi. Nilai yang diubah ialah *RXThresh_(receiver Sensitivity Threshold)*. Nilai 1.42681e-08 pada variabel tersebut memiliki artian bahwa *range* atau jangkauan yang dapat dicapai ialah sejauh 100 meter.

```

1. Phy/WirelessPhy set RXThresh_ 1.42681e-08;

```

Kode Sumber 4.6 Pengaturan *Transmission range*

```

1. set ns_      [new Simulator]
2. set tracefd  [open trace.tr w]
3. #set windowVsTime2 [open win.tr w]
4. set namtrace  [open simwrls.nam w]
5.
6.
7. $ns_ trace-all $tracefd
8. # $ns use-newtrace

```

```

9. $ns_ namtrace-all-
   wireless $namtrace $opt(x) $opt(y)
10.
11. # set up topography object
12. set topo [new Topography]
13.
14. $topo load_flatgrid $opt(x) $opt(y)
15.
16. set god_ [create-god $val(nn)]
17.
18. #
19. # Create nn mobilenodes [$val(nn)] and attach them
   to the channel.
20. #
21.
22. # configure the nodes
23.     $ns_ node-config -adhocRouting $val(rp) \
24.         -llType $val(ll) \
25.         -macType $val(mac) \
26.         -ifqType $val(ifq) \
27.         -ifqLen $val(ifqLen) \
28.         -antType $val(ant) \
29.         -propType $val(prop) \
30.         -phyType $val(netif) \
31.         -channelType $val(chan) \
32.         -topoInstance $topo \
33.         -agentTrace ON \
34.         -routerTrace ON \
35.         -macTrace OFF \
36.         -movementTrace ON
37. for {set i 0} {$i < $val(nn)} {incr i} {
38.     set node_($i) [$ns_ node]
39.     $node_($i) random-motion 0;
40. }

```

Kode Sumber 4.7 Pengaturan variabel global pada NS-2

Skrip yang ditunjukkan pada kode sumber 4.7 merupakan skrip untuk pengaturan variabel global yang diawali dengan `set ns` merupakan kode untuk pembuatan simulator baru. `set tracefd` dan `set namtrace` merupakan pengaturan untuk menentukan nama dari *trace file* dan *file network* yang akan dihasilkan dan disimpan setelah simulasi selesai dilaksanakan. `Set topo` merupakan pengaturan untuk objek topografi berdasarkan pada luas koordinat yang telah dikonfigurasi sebelumnya. `Set god` dan `node config – channelType` merupakan konfigurasi yang dilakukan pada *node-node* yang akan dibuat. Terakhir dilakukan perulangan untuk membuat pergerakan dari *node node*. *Node-node* yang dibuat tidak dapat melakukan pergerakan secara acak karena pergerakan *node* merupakan *trace file* yang dihasilkan oleh *mobility generator*.

1. `Phy/WirelessPhyExt set CSTresh_ 3.9810717055349694e-13s; # -94 dBm wireless interface sensitivity`
2. `Phy/WirelessPhyExt set Pt_ 0.002; # equals 20dBm when considering antenna gains of 1.0`
3. `Phy/WirelessPhyExt set freq_ 5.9e+9;`
4. `Phy/WirelessPhyExt set noise_floor_ 1.26e-13;# -99 dBm for 10MHz bandwidth`
5. `Phy/WirelessPhyExt set L_ 1;# default radio circuit gain/loss`
6. `Phy/WirelessPhyExt set PowerMonitorThresh_ 3.981071705534985e-18;# -174 dBm power monitor sensitivity (=level of gaussian noise)`
7. `Phy/WirelessPhyExt set HeaderDuration_ 0.000040;# 40 us`
8. `Phy/WirelessPhyExt set BasicModulationScheme_ 0;`
9. `Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1;`
10. `Phy/WirelessPhyExt set DataCaptureSwitch_ 1;`
11. `Phy/WirelessPhyExt set SINR_PreambleCapture 3.1623;# 5 dB`
12. `Phy/WirelessPhyExt set SINR_DataCapture_ 10.0;# 10 dB`
13. `Phy/WirelessPhyExt set trace_dist_ 1e6;# PHY trace until distance of 1 Mio. km ("infinity")`
14. `Phy/WirelessPhyExt set PHY_DBG_ 0;`
15. `Mac/802_11Ext set CWMin_ 15;`

```

16. Mac/802_11Ext set CWMax_ 1023;
17. Mac/802_11Ext set slotTime_ 0.000013;
18. Mac/802_11Ext set SIFS_ 0.000032;
19. Mac/802_11Ext set ShortRetryLimit_ 7;
20. Mac/802_11Ext set LongRetryLimit_ 4;
21. Mac/802_11Ext set HeaderDuration_ 0.000040;
22. Mac/802_11Ext set SymbolDuration_ 0.000008;
23. Mac/802_11Ext set BasicModulationScheme_ 0;
24. Mac/802_11Ext set use_802_11_flag_ true;
25. Mac/802_11Ext set RTSThreshold_ 2346;
26. Mac/802_11Ext set MAC_DBG 0;

```

Kode Sumber 4.8 Pengaturan IEEE 802.11p

```

1. Phy/WirelessPhyExt set CStresh_ 6.31e-12 ;#-
   82 dBm Wireless interface sensitivity ;#(sensitivit
   y defined in the standard)
2. Phy/WirelessPhyExt set Pt_ 0.001;
3. Phy/WirelessPhyExt set freq_ 5.18e+9;
4. Phy/WirelessPhyExt set noise_floor_ 2.512e-13;#-
   96 dBm for 10MHz bandwidth
5. Phy/WirelessPhyExt set L_ 1;
6. Phy/WirelessPhyExt set PowerMonitorThresh_ 1.259e-
   13;#-99dBm power monitor ;#sensitivity
7. Phy/WirelessPhyExt set HeaderDuration_ 0.000020;
8. Phy/WirelessPhyExt set BasicModulationScheme_ 0;
9. Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1;
10. Phy/WirelessPhyExt set DataCaptureSwitch_ 0;
11. Phy/WirelessPhyExt set SINR_PreambleCapture 2.5118;

12. Phy/WirelessPhyExt set SINR_DataCapture_ 100.0;
13. Phy/WirelessPhyExt set trace_dist_ 1e6; ;# PHY trac
   e until distance of 1 Mio. km; #("infinty")
14. Phy/WirelessPhyExt set PHY_DBG 0;
15. Mac/802_11Ext set CWMin_ 15;
16. Mac/802_11Ext set CWMax_ 1023;
17. Mac/802_11Ext set slotTime_ 0.000009;
18. Mac/802_11Ext set SIFS_ 0.000016;
19. Mac/802_11Ext set ShortRetryLimit_ 7;
20. Mac/802_11Ext set LongRetryLimit_ 4;
21. Mac/802_11Ext set HeaderDuration_ 0.000020;
22. Mac/802_11Ext set SymbolDuration_ 0.000004;

```

```

23. Mac/802_11Ext set BasicModulationScheme_ 0;
24. Mac/802_11Ext set use_802_11_flag_ true;
25. Mac/802_11Ext set RTSThreshold_ 2346;
26. Mac/802_11Ext set MAC_DBG 0;

```

Kode Sumber 4.9 Pengaturan IEEE 802.11a

Skrip pada kode sumber 4.8 merupakan bagian akhir dari keseluruhan skrip yang digunakan untuk menginisiasi penempatan awal *node-node* yang dibuat pada skenario *node-movement* (*mobility generation*), pergerakan *node* tersebut selama waktu simulasi dilakukan dan melakukan konfigurasi pengiriman paket data yang dilakukan nantinya dihasilkan pada *file output.tr* pada potongan skrip tersebut, akan dipanggil file skenario *node-movement* (*mobility generation*) dan *traffic-connection pattern* kemudian pengiriman paket data dimulai pada detik ke-0 dan diberhentikan pada detik ke 100 seperti yang telah di konfigurasi sebelumnya.

```

1. puts "Loading Conneciton Pattern ...."
2. source $val(cp)
3.
4. #Define traffice mode
5. puts "Loading scenarion file...."
6. source $val(sc)
7.
8. #define node initial position in nam
9.
10.   for {set i 0} {$i < $val(nn)} { incr i } {
11.       $ns_ initial_node_pos $node_($i) 20
12.   }
13.
14.
15. #tell nodes when the simulation ends
16.   for {set i 0} {$i < $val(nn)} { incr i } {
17.       $ns_ at $val(stop).0 "$node_($i) reset";
18.   }
19.

```

```

20. $ns_ at $val(stop).0002 "puts \"NS EXITING....\"";
21. $ns_ halt"
22.
23. puts $tracefd "M 0.0 nn $val(nn) x $opt(x) y $opt(y
) rp $val(rp)"
24. puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $
val(seed)"
25. puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"
26.
27. puts "Starting Simulation..."
28. $ns_ run

```

Kode Sumber 4.10 Menginisiasi penempatan awal node

4.4. Implementasi Metriks Analisis

Setelah selesai melakukan simulasi dengan NS-2 maka akan tercipta sebuah *trace file* yang berisi nilai nilai yang dapat kita gunakan untuk melakukan analisis dari simulasi yang sudah dilaksanakan. Dalam Tugas Akhir ini penulis akan melakukan analisis dari nilai rata rata *Packet Delivery Ratio*(PDR), rata rata *Routing Overhead*(RO), dan rata rata dari *End to End Delay*(E2E).

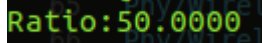
4.4.1. Implementasi *Packet Delivery Ratio*

Packet Delivery Ratio didapatkan dengan membandingkan pengiriman dan penerimaan data yang dikirimkan oleh agen. Pada Tugas Akhir ini, penulis menggunakan agen dengan pengiriman data CBR. Kemudian, pada Persamaan 2 telah dijelaskan rumus untuk menghitung *packet delivery ratio*. Skrip awk untuk menghitung *packet delivery ratio* berdasarkan kedua informasi tersebut dapat dilihat pada lampiran. Cara menjalankan skrip awk dapat dilihat pada perintah awk di bawah ini.

```
1. awk -f PDR.awk trace.tr
```

Kode Sumber 4.11 Menjalankan file PDR.awk

Hasil dari perintah yang dijalankan adalah *packet delivery ratio* dari simulasi yang telah dijalankan dapat dilihat pada gambar 4.1.



Gambar 4.1 Contoh hasil nilai PDR

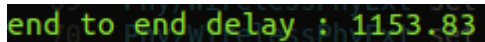
4.4.2. Implementasi *End-to-End Delay*

Perhitungan *end to end delay* didasarkan pada Persamaan 3 dan sudah dijelaskan pada bagian 3.4.2 Skrip awk untuk menghitung *end to end delay* berdasarkan kedua informasi tersebut dapat dilihat pada lampiran. Contoh perintah untuk memanggil skrip tcl untuk menganalisis *trace file* dan *outputnya* dapat dilihat pada perintah awk di bawah ini.

```
1. awk -f E2E.awk trace.tr
```

Kode Sumber 4.12 Menjalankan file E2E.awk

Hasil dari perintah yang dijalankan adalah *end-to-end delay* dari simulasi yang telah dijalankan dapat dilihat pada gambar 4.2.



Gambar 4.2 Contoh hasil nilai E2E

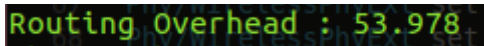
4.4.3. Implementasi *Routing Overhead*

Perhitungan RO didasarkan pada Persamaan 4 dan sudah dijelaskan pada bagian 3.4.3 Skrip awk untuk menghitung RO berdasarkan kedua informasi tersebut dapat dilihat pada lampiran. Contoh perintah untuk memanggil skrip tcl untuk menganalisis *trace file* dan *outputnya* dapat dilihat pada perintah awk di bawah ini.

```
1. awk -f ro.awk trace.tr
```

Kode Sumber 4.13 Menjalankan file ro.awk

Hasil dari perintah yang dijalankan adalah *end-to-end delay* dari simulasi yang telah dijalankan dapat dilihat pada Gambar 4.12.

A screenshot of a terminal window with a black background and green text. The text reads "Routing Overhead : 53.978".

Gambar 4.3 Contoh hasil nilai RO

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada dari skenario NS-2 yang telah dibuat. Pengujian fungsionalitas akan dibagi ke dalam beberapa skenario pengujian.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas seperti yang tertera pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Sistem

Perangkat Keras	<ul style="list-style-type: none">- Laptop Asus ROG GL502V<ul style="list-style-type: none">o Prosesor Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHzo RAM 16 GBo HDD 1 TB
Perangkat Lunak	<ul style="list-style-type: none">- Laptop Asus ROG GL502V<ul style="list-style-type: none">o Sistem Operasi Windows 10 pro, Linux Ubuntu 14.04 LTS 64 bit (NS-2, AODV, <i>Mobility Generation, Traffic Connection Genneration, 802.11p, 802.11a.</i>)

5.2. Kriteria Pengujian

Pengujian pada skenario yang dihasilkan oleh *mobility generator* default dari NS-2 menggunakan beberapa kriteria. Pada tabel 5.1 menunjukan kriteria-kriteria yang ditentukan didalam skenario.

Tabel 5.1 Penjelasan skenario pengujian

Kriteria	Spesifikasi
Skenario	MANET
Jumlah node	50,75,100

Kecepatan Maksimal Perpindahan node (m/s)	5, 10, 15
Jumlah percobaan	10
Jarak node 1 dan node 2	Acak
Posisi awal node	Acak
Pergerakan	Acak
Protokol Routing	AODV
Pengiriman Paket Data	0 – 100 Detik

5.3. Analisis *Packet Delivery Ratio (PDR)*

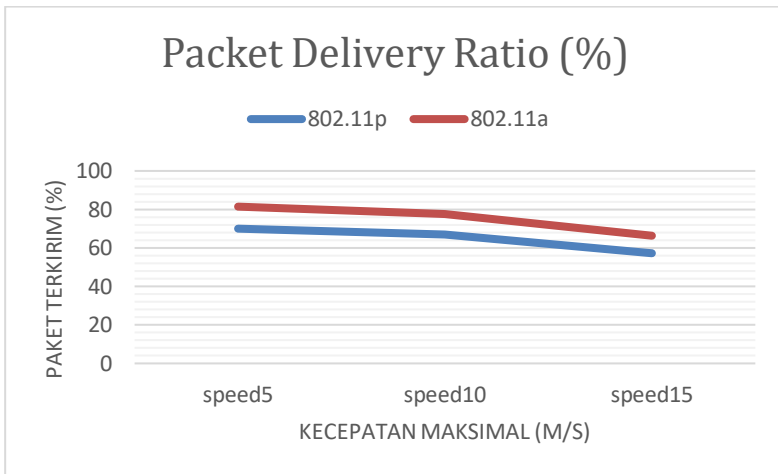
Dari hasil analisis yang dilakukan setelah melakukan uji coba berulang kali pada akhirnya penulis mendapatkan nilai rata-rata pada PDR dengan menggunakan IEEE 802.11p dan 802.11a pada *routing protocol* AODV seperti pada tabel 5.2, 5.3, 5.4, dan 5.5.

Tabel 5.2 Nilai hasil PDR 802.11p node50 statik

Kecepatan Maksimal Perpindahan Node (m/s)	PDR (%)
5	70,008
10	66,887
15	57,241

Tabel 5.3 Nilai hasil PDR 802.11a node50 statik

Kecepatan Maksimal Perpindahan Node (m/s)	PDR (%)
5	81,449
10	77,464
15	66,338



Gambar 5.1 Grafik nilai PDR 802.11p dan 802.11a dengan node static50

Pada tabel 5.2,5.3 dan gambar 5.1 menunjukkan performa PDR yang dihasilkan oleh IEEE 802.11p dan 802.11a dengan *node* yang tetap berjumlah 50 dan kecepatan maksimal yang berbeda pada *routing protocol* AODV pada jaringan MANET dengan menggunakan skenario *node-Movement (mobility generation)* yang bersifat *Random Way Point*.

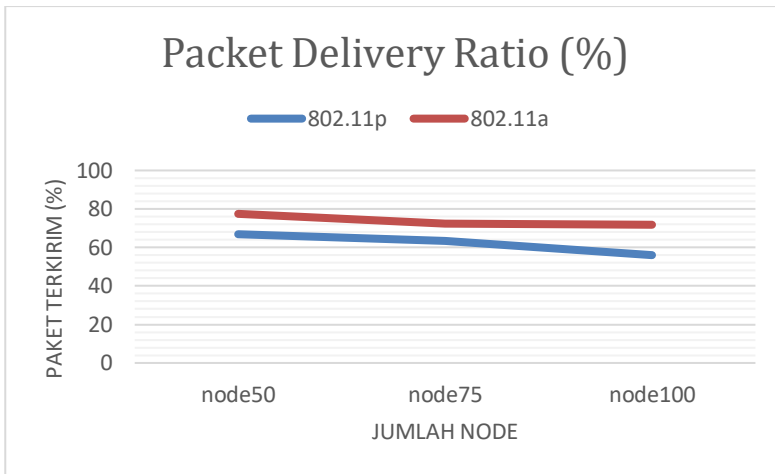
Bisa dilihat bahwa nilai rata rata dari *Packet Delivery Ratio* menunjukkan nilai yang relatif menurun. Bisa dilihat bahwa nilai rata rata pada kecepatan Maksimal 5m/s adalah 70,008% sementara pada kecepatan maksimal 10m/s memiliki nilai 66,887% dan pada kecepatan maksimal 15m/s bernilai 57,241% untuk IEEE 802.11p. Selanjutnya dapat dilihat bahwa nilai rata rata pada kecepatan Maksimal 5m/s adalah 81,449% sementara pada kecepatan maksimal 10m/s memiliki nilai 77,464% dan pada kecepatan maksimal 15m/s bernilai 66,338%. untuk IEEE 802.11a.

Tabel 5.4 Nilai hasil PDR 802.11p speed10 statik

Jumlah <i>node</i>	PDR (%)
50	66,871
75	63,305
100	56,017

Tabel 5.5 Nilai hasil PDR 802.11a speed10 statik

Jumlah <i>node</i>	PDR (%)
50	77.464
75	72,374
100	71,801

**Gambar 5.2 Grafik nilai PDR 802.11p dan 802.11a dengan speed static10**

Sama hal dengan sebelumnya pada tabel 5.4,5.5 dan gambar 5.2 menunjukan performa PDR yang dihasilkan oleh IEEE 802.11p dan 802.11a dengan kecepatan maksimal yang tetap sejumlah 10m/s dan jumlah *node* yang berbeda.

Bisa dilihat bahwa nilai rata rata pada *node* sejumlah 50 adalah 66,871% sementara pada *node* sejumlah 75 memiliki nilai

63,305% dan pada *node* sejumlah 100 bernilai 56,017% untuk IEEE 802.11p. Selanjutnya dapat dilihat bahwa nilai rata rata pada *node* sejumlah 50 adalah 77,464% sementara pada *node* sejumlah 75 memiliki nilai 72,374% dan pada *node* sejumlah 100 bernilai 71,801%. untuk IEEE 802.11a.

Hal ini menunjukkan bahwa semakin tinggi kecepatan maksimal yang dipunya maka semakin rendah pula paket data yang dikirimkan. Pergerakan *node* yang lebih dinamis inilah yang memungkinkan terjadinya banyak rute putus saat pengiriman paket data ataupun kegagalan pembentukan tabel *routing* yang dibuat oleh AODV sehingga menyebabkan paket data yang dikirimkan tidak sampai ke tujuan. Selain itu juga dapat dilihat bahwa IEEE 802.11a mengirimkan jumlah paket lebih banyak dari IEEE 802.11p, Selain itu juga jumlah *node* yang di inputkan dapat mempengaruhi persentase paket data yang dikirimkan. Hal lain yang dapat mempengaruhi penurunan ataupun peningkatan nilai PDR yang dihasilkan adalah penempatan dan pergerakan secara acak yang di implementasikan pada skenario yang dihasilkan oleh file *node-movement (mobility generation)*.

5.4. Analisis Routing Overhead(RO)

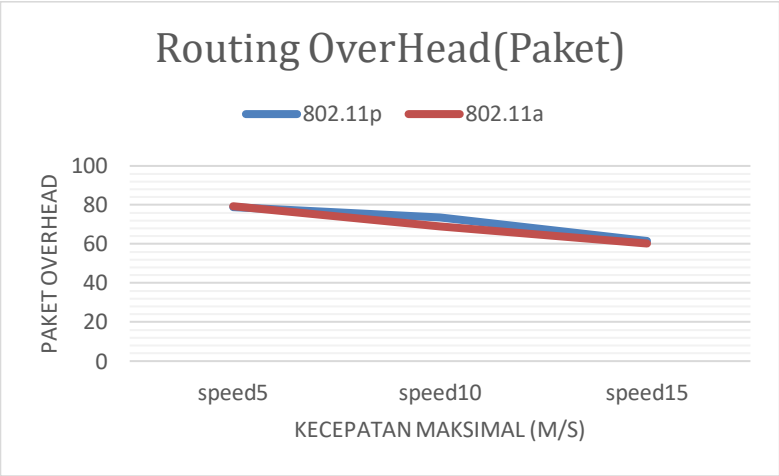
Dari hasil analisis yang dilakukan setelah melakukan uji coba berulang kali pada akhir penulis mendapatkan nilai rata-rata pada RO dengan menggunakan IEEE 802.11p dan 802.11a seperti pada tabel 5.6, 5.7, 5.8, dan 5.9.

Tabel 5.6 Nilai hasil RO 802.11p node50 statik

Kecepatan Maksimal Perpindahan Node (m/s)	RO (paket)
5	78,878
10	73,738
15	61,563

Tabel 5.7 Nilai hasil RO 802.11a node50 statik

Kecepatan Maksimal Perpindahan Node (m/s)	RO (paket)
5	79,348
10	69,134
15	60,248



Gambar 5.3 Grafik nilai RO 802.11p dan 802.11a dengan node statik50

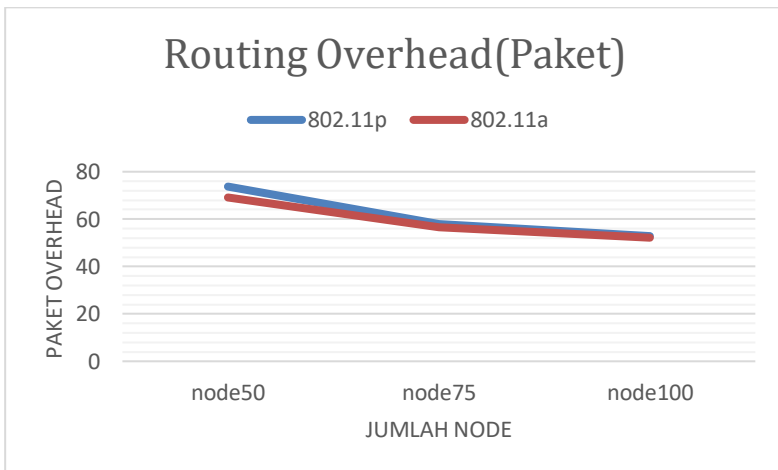
Pada gambar 5.3 dan tabel 5.6 berserta 5.7 dari hasil skenario yang sudah dibuat bisa dilihat bahwa nilai rata rata dari *Routing Overhead* yang dihasilkan bersifat relatif menurun. Untuk IEEE 802.11p pada kecepatan maksimal 5m/s adalah 78,878 paket sementara pada kecepatan maksimal 10m/s adalah 73,738 paket dan pada kecepatan maksimal 15m/s adalah 61,563 paket. Sedangkan Untuk IEEE 802.11a pada kecepatan maksimal 5m/s adalah 79,348 paket sementara pada kecepatan maksimal 10m/s adalah 69,134 paket dan pada kecepatan maksimal 15m/s adalah 60,248 paket.

Tabel 5.8 Nilai hasil RO 802.11p speed10 statik

Jumlah <i>Node</i>	RO (paket)
50	73,738
75	57,914
100	52,830

Tabel 5.9 Nilai hasil RO 802.11a node50 statik

Jumlah <i>Node</i>	RO (paket)
50	69,134
75	56,634
100	52,155

**Gambar 5.4 Grafik nilai RO 802.11p dan 802.11a dengan speed statik10**

Pada gambar 5.3 dan tabel 5.8 berserta 5.9 dari hasil skenario yang sudah dibuat bisa dilihat bahwa nilai rata rata dari *Routing Overhead* yang dihasilkan bersifat relatif menurun. Untuk IEEE 802.11p pada *node* sejumlah 50 adalah 73,738 paket sementara pada *node* sejumlah 75 adalah 57,914 paket dan pada *node* sejumlah 100 adalah 52,830 paket. Sedangkan Untuk IEEE 802.11a pada *node* sejumlah 50 adalah 69,134 paket sementara

pada *node* sejumlah 75 adalah 56,634 paket dan pada *node* sejumlah 100 adalah 52,155 paket.

Dapat dilihat pada kedua gambar grafik 5.3 dan 5.4, pada kedua grafik dapat dilihat bahwa hasil 802.11a cenderung lebih baik dari pada 802.11p. Hal ini dapat terjadi akibat mobilitas pengiriman paket antar *node* yang sangat dinamis pada *skenario node-movement(mobility generation)* yang dihasilkan oleh IEEE 802.11p dan 802.11a. Mobilitas yang sangat dinamis ini memungkinkan terjadinya rute putus saat pengiriman paket data sehingga pengiriman paket dimasukkan ke dalam antrian dan menunggu rute baru terbentuk sebelum pengiriman paket data dilanjutkan kembali.

5.5. Analisis *End-to-End Delay (E2E)*

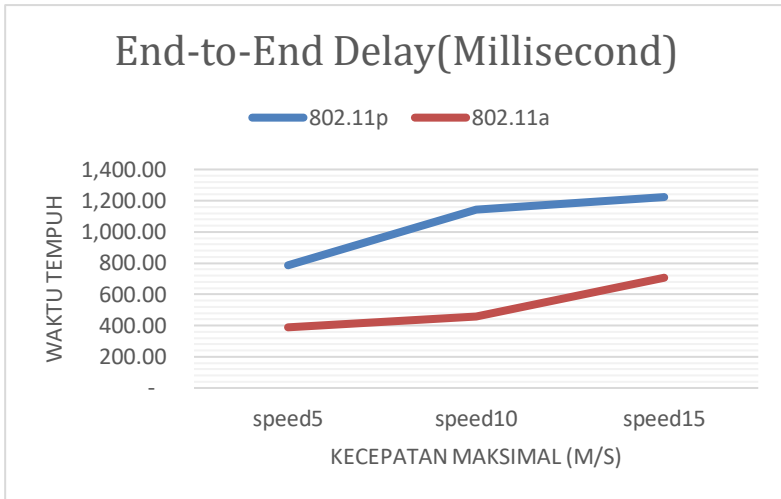
Dari hasil analisis yang dilakukan setelah melakukan uji coba berulang kali pada akhir penulis mendapatkan nilai rata-rata pada E2E dengan menggunakan IEEE 802.11p dan 802.11a menjadi tabel 5.10, 5.11, 5.12, dan 5.13.

Tabel 5.10 Hasil nilai E2E 802.11p node50 statik

Kecepatan Maksimal Perpindahan Node (m/s)	End to End delay (ms)
5	786,012
10	1.141,923
15	1.223,125

Tabel 5.11 Hasil nilai E2E 802.11a node50 statik

Kecepatan Maksimal Perpindahan Node (m/s)	End to End delay (ms)
5	388,529
10	459,613
15	706,377



Gambar 5.5 Grafik hasil nilai E2E dengan node statik50

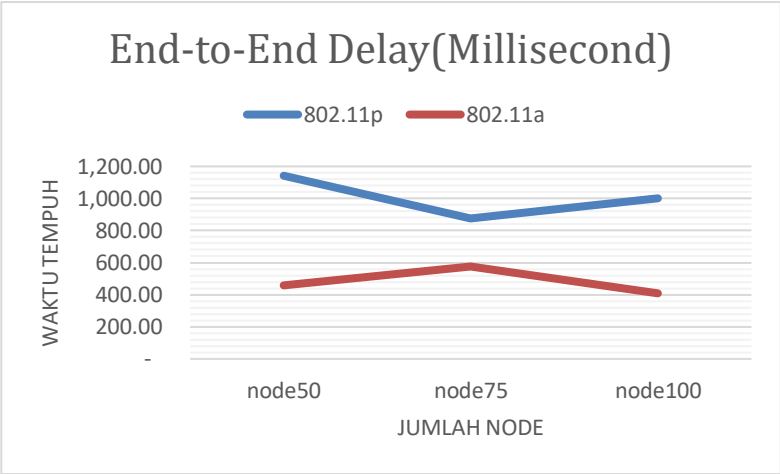
Pada tabel 5.10 dan 5.11 dengan gambar 5.5 menunjukkan pengujian IEEE 802.11p dan 802.11a untuk nilai rata rata dari *End to End Delay* bersifat relatif meningkat pada jumlah waktu yang dibutuhkan untuk mengirimkan paket data yang dikirimkan. Pada IEEE 802.11p kecepatan maksimal 5m/s rata rata nilainya adalah 786,012 ms sementara pada kecepatan maksimal 10m/s adalah 1.141,923 ms dan pada kecepatan maksimal 15m/s adalah 1.223,125 ms. Sedangkan pada IEEE 802.11a kecepatan maksimal 5m/s rata rata nilainya adalah 388,529 ms sementara pada kecepatan maksimal 10m/s adalah 459,613 ms dan pada kecepatan maksimal 15m/s adalah 706,377 ms. Hal ini terjadi dikarenakan kecepatan yang berubah menjadi semakin cepat mulai dari kecepatan maksimum 5m/s menjadi 10m/s dan 15m/s.

Tabel 5.12 Hasil nilai E2E 802.11p speed10 statik

Jumlah <i>Node</i>	End to End delay (ms)
50	1.141,923
75	875,558
100	1.001,326

Tabel 5.13 Hasil nilai E2E 802.11a speed10 statik

Jumlah <i>Node</i>	End to End delay (ms)
50	459,613
75	575,945
100	408,765



Gambar 5.6 Grafik hasil nilai E2E dengan speed statik10

Pada tabel 5.10 dan 5.11 dengan gambar 5.5 menunjukkan pengujian IEEE 802.11p dan 802.11a untuk nilai rata rata dari *End to End Delay* bersifat fluktuatif pada jumlah waktu yang dibutuhkan untuk mengirimkan paket data yang dikirimkan. Pada IEEE 802.11p *node* sejumlah 50 rata rata nilainya adalah 1.141,923 ms, sementara pada *node* sejumlah 75 adalah 875,558 ms dan pada *node* sejumlah 100 adalah 1.001,326 ms. Sedangkan pada IEEE 802.11a *node*

sejumlah 50 rata rata nilainya adalah 459,613 ms sementara *node* sejumlah 75 adalah 575,945 ms dan pada *node* sejumlah 100 adalah 408,765 ms. Hal ini terjadi dikarenakan kecepatan yang stabil dan bergantinya jumlah *node*.

Pada gambar 5.6 dapat dilihat terjadi fluktuatif pada waktu tempuh kedua IEEE 802.11p dan 802.11a terjadi fluktuatif ini bisa disebabkan berbagai macam hal seperti *transmission delay*, *propagation delay*, *processing delay*, dan *queuing delay*. Hal ini terjadinya pada saat *buffer* pada *route discovery latency, delay* pada antrian *interface*, retransmisi.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Kesimpulan yang dapat diambil dalam tugas akhir ini adalah sebagai berikut:

1. Berikut ini merupakan nilai rata-rata dari PDR, RO, E2E pada MANET menggunakan protokol *routing* AODV.
 - Nilai rata-rata PDR dengan jumlah *node* statik yang didapat pada IEEE 802.11p adalah 64,707% dan IEEE 802.11a adalah 75,084%.
 - Nilai rata-rata PDR dengan kecepatan maksimal statik yang didapat pada IEEE 802.11p adalah 62,064% dan IEEE 802.11a adalah 73,880%.
 - Nilai rata-rata RO dengan jumlah *node* statik yang didapat pada IEEE 802.11p adalah 71,393 paket yang terkirim dan IEEE 802.11a adalah 69,577 paket yang terkirim.
 - Nilai rata-rata RO dengan kecepatan maksimal statik yang didapat pada IEEE 802.11p adalah 61,404 paket yang terkirim dan IEEE 802.11a adalah 69,577 paket yang terkirim.
 - Nilai rata-rata E2E dengan jumlah *node* statik yang didapat pada IEEE 802.11p adalah 1.050,354 ms dan IEEE 802.11a adalah 518,173 ms untuk mengirimkan paket.
 - Nilai rata-rata E2E dengan kecepatan maksimal statik yang didapat pada IEEE 802.11p adalah

1.006,269 ms dan IEEE 802.11a adalah 481,441 ms untuk mengirimkan paket.

2. Berikut ini merupakan kesimpulan dari analisis IEEE 802.11p dan 802.11a pada routing protocol AODV di lingkungan MANET

- Pada nilai rata-rata PDR kedua grafik mengalami penurunan, dari hasil grafik dapat dilihat IEEE 802.11a memiliki persentase lebih baik dibandingkan IEEE 802.11p pada grafik kecepatan dan node.
- Pada nilai rata-rata RO kedua grafik mengalami penurunan pada grafik kecepatan 802.11a memiliki hasil *routing overhead* yang lebih kecil dibandingkan 802.11p untuk kedua grafik, tetapi hasilnya semakin mendekati sama pada *node* yang lebih besar.
- Pada nilai rata-rata E2E kedua grafik terjadi perubahan yang cukup signifikan dan fluktuatif, pada grafik kecepatan 802.11p memiliki waktu yang lebih dibandingkan dengan 802.11a pada kecepatan 5m/s dan 10m/s 802.11p terjadi kenaikan yang signifikan sedangkan pada 10m/s dan 15m/s 802.11a terjadi kenaikan yang signifikan. Sedangkan pada grafik node terjadi kenaikan dan penurunan waktu pada kedua IEEE 802.11p dan 802.11a.

6.2. Saran

Dalam pengerjaan Tugas Akhir ini terdapat beberapa saran untuk perbaikan serta pengembangan sistem yang telah dikerjakan sebagai berikut:

1. Dapat dilakukan dengan penambahan jumlah *node* diatas 100 karena terdapat hasil yang lebih unik dan penambahan jumlah percobaan untuk skenario *node-movement(mobility generation)* untuk hasil yang lebih akurat.

2. Dapat dilakukan modifikasi pada parameter parameter lain yang berhubungan dengan simulasi AODV di NS2.
3. Dapat melakukan perbandingan dengan IEEE 802.11 yang lain sehingga mendapatkan hasil yang lebih baik.
4. Dapat menggunakan propagasi sebagai acuan untuk menganalisis hasil yang berbeda.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] P. C. Sekhar, M. R. P. Kumar, B. P. Kumar, and C. Koteswararao, "Impact of Routing Overhead in A Real-Time MANET Environment," vol. 4, p. 7, 2013.
- [2] "MANET: Mobile Ad hoc Network," *GeeksforGeeks*, 08-Mar-2018. .
- [3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," Jul. 2003.
- [4] S. Eichler, "Performance Evaluation of the IEEE 802.11p WAVE Communication Standard," in *2007 IEEE 66th Vehicular Technology Conference*, 2007, pp. 2199–2203.
- [5] S. Banerji and R. Singha Chowdhury, "On IEEE 802.11: Wireless Lan Technology," *Int. J. Mob. Netw. Commun. Telemat.*, vol. 3, Jul. 2013.
- [6] "The Network Simulator - ns-2." [Online]. Available: <https://www.isi.edu/nsnam/ns/>. [Accessed: 12-Jan-2019].
- [7] "AWK, oh AWK - Mufid's Code blog." [Online]. Available: <https://mufid.github.io/blog/2012/awk/>. [Accessed: 12-Jan-2019].
- [8] D. A. Maltz, "On-Demand Routing in Multi-hop Wireless Mobile Ad Hoc Networks:," Defense Technical Information Center, Fort Belvoir, VA, May 2001.
- [9] R. Kumar, P. Verma, and Y. Singh, "Mobile Ad Hoc Networks and It's Routing Protocols," vol. 7, no. 8, p. 10, 2013.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

```
1. #
2. # nodes: 50, pause: 10.00, max speed: 5.00, max x:
   800.00, max y: 800.00
3. #
4. $node_(0) set X_ 697.903685915276
5. $node_(0) set Y_ 513.159406636180
6. $node_(0) set Z_ 0.000000000000
7. $node_(1) set X_ 438.373617823634
8. $node_(1) set Y_ 129.472095083957
9. $node_(1) set Z_ 0.000000000000
10. $node_(2) set X_ 536.865315509025
11. $node_(2) set Y_ 558.214452888026
12. $node_(2) set Z_ 0.000000000000
13. $node_(3) set X_ 179.289688284568
14. $node_(3) set Y_ 5.433852116168
15. $node_(3) set Z_ 0.000000000000
16. $node_(4) set X_ 448.633979381031
17. $node_(4) set Y_ 4.965617003117
18. $node_(4) set Z_ 0.000000000000
19. $node_(5) set X_ 784.455524190969
20. $node_(5) set Y_ 518.186644972959
21. $node_(5) set Z_ 0.000000000000
22. $node_(6) set X_ 708.710454081133
23. $node_(6) set Y_ 792.490677544396
24. $node_(6) set Z_ 0.000000000000
25. $node_(7) set X_ 545.587415965782
26. $node_(7) set Y_ 516.862728704570
27. $node_(7) set Z_ 0.000000000000
28. $node_(8) set X_ 270.475344047619
29. $node_(8) set Y_ 668.388120044193
30. $node_(8) set Z_ 0.000000000000
31. $node_(9) set X_ 258.323682316422
32. $node_(9) set Y_ 229.786260564705
33. $node_(9) set Z_ 0.000000000000
34. $node_(10) set X_ 299.993538669929
35. $node_(10) set Y_ 240.424680246118
36. $node_(10) set Z_ 0.000000000000
37. $node_(11) set X_ 613.273793729243
38. $node_(11) set Y_ 634.041180806366
39. $node_(11) set Z_ 0.000000000000
```

```

40. $node_(12) set X_ 621.534272953665
41. $node_(12) set Y_ 287.403848343529
42. $node_(12) set Z_ 0.000000000000
43. $node_(13) set X_ 400.748524664830
44. $node_(13) set Y_ 476.018143587693
45. $node_(13) set Z_ 0.000000000000
46. $node_(14) set X_ 239.627692718655
47. $node_(14) set Y_ 294.458669714491
48. $node_(14) set Z_ 0.000000000000
49. $node_(15) set X_ 203.090852462430
50. $node_(15) set Y_ 177.793255490483
51. $node_(15) set Z_ 0.000000000000
52. $node_(16) set X_ 799.390476726280
53. $node_(16) set Y_ 431.789911774057
54. $node_(16) set Z_ 0.000000000000
55. $node_(17) set X_ 290.609825506537
56. $node_(17) set Y_ 44.194068851035
57. $node_(17) set Z_ 0.000000000000
58. $node_(18) set X_ 559.684546388310
59. $node_(18) set Y_ 412.649584056603
60. $node_(18) set Z_ 0.000000000000
61. $node_(19) set X_ 559.350566087504
62. $node_(19) set Y_ 169.658829478788
63. $node_(19) set Z_ 0.000000000000
64. $node_(20) set X_ 674.044616381005
65. $node_(20) set Y_ 384.732167242163
66. $node_(20) set Z_ 0.000000000000
67. $node_(21) set X_ 336.686311203696
68. $node_(21) set Y_ 449.919972192346
69. $node_(21) set Z_ 0.000000000000
70. $node_(22) set X_ 739.489857529730
71. $node_(22) set Y_ 262.000189557681
72. $node_(22) set Z_ 0.000000000000
73. $node_(23) set X_ 311.217126607234
74. $node_(23) set Y_ 546.515038626997
75. $node_(23) set Z_ 0.000000000000
76. $node_(24) set X_ 168.270160956353
77. $node_(24) set Y_ 12.090419259576
78. $node_(24) set Z_ 0.000000000000
79. $node_(25) set X_ 600.410579164839
80. $node_(25) set Y_ 429.859215722121
81. $node_(25) set Z_ 0.000000000000
82. $node_(26) set X_ 752.166834951076

```

```

83. $node_(26) set Y_ 784.922249816318
84. $node_(26) set Z_ 0.000000000000
85. $node_(27) set X_ 378.684275868891
86. $node_(27) set Y_ 441.074332162612
87. $node_(27) set Z_ 0.000000000000
88. $node_(28) set X_ 436.107866352060
89. $node_(28) set Y_ 387.060733816734
90. $node_(28) set Z_ 0.000000000000
91. $node_(29) set X_ 75.549413197273
92. $node_(29) set Y_ 554.310499712961
93. $node_(29) set Z_ 0.000000000000
94. $node_(30) set X_ 608.667215938396
95. $node_(30) set Y_ 246.350320438125
96. $node_(30) set Z_ 0.000000000000
97. $node_(31) set X_ 12.274515664461
98. $node_(31) set Y_ 785.431431273403
99. $node_(31) set Z_ 0.000000000000

```

Kode Sumber 7.1 Posisi *node* dari potongan Skenario

```

1. 1. $god_ set-dist 0 1 2
2. 2. $god_ set-dist 0 2 1
3. 3. $god_ set-dist 0 3 4
4. 4. $god_ set-dist 0 4 3
5. 5. $god_ set-dist 0 5 1
6. 6. $god_ set-dist 0 6 2
7. 7. $god_ set-dist 0 7 1
8. 8. $god_ set-dist 0 8 3
9. 9. $god_ set-dist 0 9 3
10. 10. $god_ set-dist 0 10 3
11. 11. $god_ set-dist 0 11 1
12. 12. $god_ set-dist 0 12 1
13. 13. $god_ set-dist 0 13 2
14. 14. $god_ set-dist 0 14 3
15. 15. $god_ set-dist 0 15 3
16. 16. $god_ set-dist 0 16 1
17. 17. $god_ set-dist 0 17 3
18. 18. $god_ set-dist 0 18 1
19. 19. $god_ set-dist 0 19 2
20. 20. $god_ set-dist 0 20 1
21. 21. $god_ set-dist 0 21 2
22. 22. $god_ set-dist 0 22 2

```

```
23. 23. $god_ set-dist 0 23 2
24. 24. $god_ set-dist 0 24 4
25. 25. $god_ set-dist 0 25 1
26. 26. $god_ set-dist 0 26 2
27. 27. $god_ set-dist 0 27 2
28. 28. $god_ set-dist 0 28 2
29. 29. $god_ set-dist 0 29 3
30. 30. $god_ set-dist 0 30 2
31. 31. $god_ set-dist 0 31 4
32. 32. $god_ set-dist 0 32 4
33. 33. $god_ set-dist 0 33 1
34. 34. $god_ set-dist 0 34 3
35. 35. $god_ set-dist 0 35 4
36. 36. $god_ set-dist 0 36 1
37. 37. $god_ set-dist 0 37 3
38. 38. $god_ set-dist 0 38 3
39. 39. $god_ set-dist 0 39 3
40. 40. $god_ set-dist 0 40 2
41. 41. $god_ set-dist 0 41 2
42. 42. $god_ set-dist 0 42 1
43. 43. $god_ set-dist 0 43 2
44. 44. $god_ set-dist 0 44 4
45. 45. $god_ set-dist 0 45 2
46. 46. $god_ set-dist 0 46 2
47. 47. $god_ set-dist 0 47 3
48. 48. $god_ set-dist 0 48 1
49. 49. $god_ set-dist 0 49 3
50. 50. $god_ set-dist 1 2 3
51. 51. $god_ set-dist 1 3 2
52. 52. $god_ set-dist 1 4 1
53. 53. $god_ set-dist 1 5 3
54. 54. $god_ set-dist 1 6 4
55. 55. $god_ set-dist 1 7 2
56. 56. $god_ set-dist 1 8 3
57. 57. $god_ set-dist 1 9 1
58. 58. $god_ set-dist 1 10 1
59. 59. $god_ set-dist 1 11 3
60. 60. $god_ set-dist 1 12 1
61. 61. $god_ set-dist 1 13 2
62. 62. $god_ set-dist 1 14 2
63. 63. $god_ set-dist 1 15 1
64. 64. $god_ set-dist 1 16 2
65. 65. $god_ set-dist 1 17 1
```

```

66. 66. $god_ set-dist 1 18 2
67. 67. $god_ set-dist 1 19 1
68. 68. $god_ set-dist 1 20 2
69. 69. $god_ set-dist 1 21 2
70. 70. $god_ set-dist 1 22 2
71. 71. $god_ set-dist 1 23 2
72. 72. $god_ set-dist 1 24 2
73. 73. $god_ set-dist 1 25 2
74. 74. $god_ set-dist 1 26 4
75. 75. $god_ set-dist 1 27 2
76. 76. $god_ set-dist 1 28 2
77. 77. $god_ set-dist 1 29 3
78. 78. $god_ set-dist 1 30 1
79. 79. $god_ set-dist 1 31 4
80. 80. $god_ set-dist 1 32 2
81. 81. $god_ set-dist 1 33 3
82. 82. $god_ set-dist 1 34 3
83. 83. $god_ set-dist 1 35 2
84. 84. $god_ set-dist 1 36 3
85. 85. $god_ set-dist 1 37 1
86. 86. $god_ set-dist 1 38 3
87. 87. $god_ set-dist 1 39 3
88. 88. $god_ set-dist 1 40 1
89. 89. $god_ set-dist 1 41 2
90. 90. $god_ set-dist 1 42 1
91. 91. $god_ set-dist 1 43 4
92. 92. $god_ set-dist 1 44 2
93. 93. $god_ set-dist 1 45 1
94. 94. $god_ set-dist 1 46 3
95. 95. $god_ set-dist 1 47 2
96. 96. $god_ set-dist 1 48 3
97. 97. $god_ set-dist 1 49 3

```

Kode Sumber 7.2 Pembuatan ‘GOD’ dari potongan skenario

```

1. #
2. # nodes: 2, max conn: 1, send rate: 4.0, seed: 1.0
3. #
4. #
5. # 1 connecting to 2 at time 2.5568388786897245
6. #
7. set udp_(0) [new Agent/UDP]
8. $ns_ attach-agent $node_(1) $udp_(0)
9. set null_(0) [new Agent/Null]
10. $ns_ attach-agent $node_(2) $null_(0)
11. set cbr_(0) [new Application/Traffic/CBR]
12. $cbr_(0) set packetSize_ 512
13. $cbr_(0) set interval_ 4.0
14. $cbr_(0) set random_ 1
15. $cbr_(0) set maxpkts_ 10000
16. $cbr_(0) attach-agent $udp_(0)
17. $ns_ connect $udp_(0) $null_(0)
18. $ns_ at 2.5568388786897245 "$cbr_(0) start"
19. #
20. #Total sources/connections: 1/1
21. #

```

Kode Sumber 7.3 Koneksi yang digunakan pada ‘cbr.txt’

```

1. # A 10-node example for ad-
   hoc simulation with AODV
2.
3. # Define options
4. set val(chan)          Channel/WirelessChannel
   ;# channel type
5. set val(prop)          Propagation/TwoRayGround
   ;# radio-propagation model
6. set val(netif)         Phy/WirelessPhyExt
   ;# network interface type
7. set val(mac)           Mac/802_11Ext
   ;# MAC type
8. set val(ifq)           Queue/DropTail/PriQueue
   ;# interface queue type
9. set val(ll)            LL
   ;# link layer type

```

```

10. set val(ant)           Antenna/OmniAntenna
    ;# antenna model
11. set val(ifqlen)        50
    ;# max packet in ifq
12. set val(nn)            100
    ;# number of mobilenodes
13. set val(rp)            AODV
    ;# routing protocol
14. set opt(x)             800                ;# X dim
    ension of topography
15. set opt(y)             800                ;# Y dim
    ension of topography
16. set val(stop)         100                ;# time of s
    imulation end
17. set val(seed)          0                  ;
18. set val(cp)            "cbr.txt";
19. set val(sc)            "scenario.txt";
20.
21. Phy/WirelessPhy set RXThresh_ 1.42681e-08;
22.
23. set ns_                [new Simulator]
24. set tracefd            [open trace.tr w]
25. #set windowVsTime2    [open win.tr w]
26. set namtrace           [open simwrls.nam w]
27.
28.
29. $ns_ trace-all $tracefd
30. # $ns use-newtrace
31. $ns_ namtrace-all-
    wireless $namtrace $opt(x) $opt(y)
32.
33. # set up topography object
34. set topo               [new Topography]
35.
36. $topo load_flatgrid $opt(x) $opt(y)
37.
38. set god_ [create-god $val(nn)]
39.
40.
41.
42.
43. #

```

```

44. # Create nn mobilenodes [$val(nn)] and attach them
    to the channel.
45. #
46.
47. # configure the nodes
48.     $ns_ node-config -adhocRouting $val(rp) \
49.         -llType $val(ll) \
50.         -macType $val(mac) \
51.         -ifqType $val(ifq) \
52.         -ifqLen $val(ifqlen) \
53.         -antType $val(ant) \
54.         -propType $val(prop) \
55.         -phyType $val(netif) \
56.         -channelType $val(chan) \
57.         -topoInstance $topo \
58.         -agentTrace ON \
59.         -routerTrace ON \
60.         -macTrace OFF \
61.         -movementTrace ON
62.
63. Phy/WirelessPhyExt set CSTresh_ 3.9810717055349694e
    -13s;
64. Phy/WirelessPhyExt set Pt_ 0.002;
65. Phy/WirelessPhyExt set freq_ 5.9e+9;
66. Phy/WirelessPhyExt set noise_floor_ 1.26e-13;
67. Phy/WirelessPhyExt set L_ 1;
68. Phy/WirelessPhyExt set PowerMonitorThresh_ 3.981071
    705534985e-18;
69. Phy/WirelessPhyExt set HeaderDuration_ 0.000040;
70. Phy/WirelessPhyExt set BasicModulationScheme_ 0;
71. Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1;
72. Phy/WirelessPhyExt set DataCaptureSwitch_ 1;
73. Phy/WirelessPhyExt set SINR_PreambleCapture 3.1623;

74. Phy/WirelessPhyExt set SINR_DataCapture_ 10.0;
75. Phy/WirelessPhyExt set trace_dist_ 1e6;
76. Phy/WirelessPhyExt set PHY_DBG_ 0;
77. Mac/802_11Ext set CWMin_ 15;
78. Mac/802_11Ext set CWMax_ 1023;
79. Mac/802_11Ext set slotTime_ 0.000013;
80. Mac/802_11Ext set SIFS_ 0.000032;
81. Mac/802_11Ext set ShortRetryLimit_ 7;
82. Mac/802_11Ext set LongRetryLimit_ 4;

```



```

83. Mac/802_11Ext set HeaderDuration_ 0.000040;
84. Mac/802_11Ext set SymbolDuration_ 0.000008;
85. Mac/802_11Ext set BasicModulationScheme_ 0;
86. Mac/802_11Ext set use_802_11_flag_ true;
87. Mac/802_11Ext set RTSThreshold_ 2346;
88. Mac/802_11Ext set MAC_DBG 0;
89.
90.     for {set i 0} {$i < $val(nn) } { incr i } {
91.         set node_($i) [$ns_ node]
92.         $node_($i) random-motion 0;
93.     }
94.
95.
96.
97. # Define node movement model
98. puts "Loading Conneciton Pattern ...."
99. source $val(cp)
100.
101.     #Define traffice mode
102.     puts "Loading scenarion file...."
103.     source $val(sc)
104.
105.     #define node initial position in nam
106.
107.         for {set i 0} {$i < $val(nn) } { incr i
108.             } {
109.                 $ns_ initial_node_pos $node_($i) 20
110.             }
111.
112.     #tell nodes when the simulation ends
113.     for {set i 0} {$i < $val(nn) } { incr i
114.         } {
115.             $ns_ at $val(stop).0 "$node_($i) res
116.             et";
117.         }
118.         $ns_ at $val(stop).0002 "puts \"NS EXITING..
119.         ..\"";
120.         $ns_ halt"
121.

```

```

120.      puts $tracefd "M 0.0 nn $val(nn) x $opt(x) y
      $opt(y) rp $val(rp)"
121.      puts $tracefd "M 0.0 sc $val(sc) cp $val(cp)
      seed $val(seed)"
122.      puts $tracefd "M 0.0 prop $val(prop) ant $va
      l(ant)"
123.
124.      puts "Starting Simulation..."
125.      $ns_ run

```

Kode Sumber 7.4 file .tcl untuk simulasi AODV 802.11p

```

1.  # A 10-node example for ad-
    hoc simulation with AODV
2.
3.  # Define options
4.  set val(chan)          Channel/WirelessChannel
    ;# channel type
5.  set val(prop)          Propagation/TwoRayGround
    ;# radio-propagation model
6.  set val(netif)         Phy/WirelessPhyExt
    ;# network interface type
7.  set val(mac)           Mac/802_11Ext
    ;# MAC type
8.  set val(ifq)           Queue/DropTail/PriQueue
    ;# interface queue type
9.  set val(ll)            LL
    ;# link layer type
10. set val(ant)           Antenna/OmniAntenna
    ;# antenna model
11. set val(ifqlen)        50
    ;# max packet in ifq
12. set val(nn)            100
    ;# number of mobilenodes
13. set val(rp)            AODV
    ;# routing protocol
14. set opt(x)             800                ;# X dim
    ension of topography
15. set opt(y)             800                ;# Y dim
    ension of topography

```

```

16. set val(stop)          100          ;# time of s
    imulation end
17. set val(seed)          0            ;
18. set val(cp)             "cbr.txt";
19. set val(sc)             "scenario.txt";
20.
21. Phy/WirelessPhy set RXThresh_ 1.42681e-08;
22.
23. set ns_                 [new Simulator]
24. set tracefd             [open trace.tr w]
25. #set windowVsTime2     [open win.tr w]
26. set namtrace            [open simwrls.nam w]
27.
28.
29. $ns_ trace-all $tracefd
30. #$ns_ use-newtrace
31. $ns_ namtrace-all-
    wireless $namtrace $opt(x) $opt(y)
32.
33. # set up topography object
34. set topo                [new Topography]
35.
36. $topo load_flatgrid $opt(x) $opt(y)
37.
38. set god_ [create-god $val(nn)]
39.
40.
41.
42.
43. #
44. # Create nn mobilenodes [$val(nn)] and attach them
    to the channel.
45. #
46.
47. # configure the nodes
48.     $ns_ node-config -adhocRouting $val(rp) \
49.         -llType $val(ll) \
50.         -macType $val(mac) \
51.         -ifqType $val(ifq) \
52.         -ifqLen $val(ifqlen) \
53.         -antType $val(ant) \
54.         -propType $val(prop) \
55.         -phyType $val(netif) \

```

```

56.             -channelType $val(chan) \
57.             -topoInstance $topo \
58.             -agentTrace ON \
59.             -routerTrace ON \
60.             -macTrace OFF \
61.             -movementTrace ON
62.
63. Phy/WirelessPhyExt set CStresh_ 6.31e-12 ;
64. Phy/WirelessPhyExt set Pt_ 0.001;
65. Phy/WirelessPhyExt set freq_ 5.18e+9;
66. Phy/WirelessPhyExt set noise_floor_ 2.512e-13;
67. Phy/WirelessPhyExt set L_ 1;
68. Phy/WirelessPhyExt set PowerMonitorThresh_ 1.259e-
    13;
69. Phy/WirelessPhyExt set HeaderDuration_ 0.000020;
70. Phy/WirelessPhyExt set BasicModulationScheme_ 0;
71. Phy/WirelessPhyExt set PreambleCaptureSwitch_ 1;
72. Phy/WirelessPhyExt set DataCaptureSwitch_ 0;
73. Phy/WirelessPhyExt set SINR_PreambleCapture 2.5118;

74. Phy/WirelessPhyExt set SINR_DataCapture_ 100.0;
75. Phy/WirelessPhyExt set trace_dist_ 1e6;
76. Phy/WirelessPhyExt set PHY_DBG_ 0;
77. Mac/802_11Ext set CWMin_ 15;
78. Mac/802_11Ext set CWMax_ 1023;
79. Mac/802_11Ext set slotTime_ 0.000009;
80. Mac/802_11Ext set SIFS_ 0.000016;
81. Mac/802_11Ext set ShortRetryLimit_ 7;
82. Mac/802_11Ext set LongRetryLimit_ 4;
83. Mac/802_11Ext set HeaderDuration_ 0.000020;
84. Mac/802_11Ext set SymbolDuration_ 0.000004;
85. Mac/802_11Ext set BasicModulationScheme_ 0;
86. Mac/802_11Ext set use_802_11_flag_ true;
87. Mac/802_11Ext set RTSThreshold_ 2346;
88. Mac/802_11Ext set MAC_DBG 0;
89.
90.     for {set i 0} {$i < $val(nn)} {incr i} {
91.         set node_($i) [$ns_ node]
92.         $node_($i) random-motion 0;
93.     }
94.
95.
96.

```

```

97. # Define node movement model
98. puts "Loading Conneciton Pattern ...."
99. source $val(cp)
100.
101.     #Define traffice mode
102.     puts "Loading scenarion file...."
103.     source $val(sc)
104.
105.     #define node initial position in nam
106.
107.         for {set i 0} {$i < $val(nn) } { incr i
108.             {
109.                 $ns_ initial_node_pos $node_($i) 20
110.             }
111.
112.         #tell nodes when the simulation ends
113.         for {set i 0} {$i < $val(nn) } { incr i
114.             {
115.                 $ns_ at $val(stop).0 "$node_($i) res
116.                 et";
117.             }
118.
119.             $ns_ at $val(stop).0002 "puts \"NS EXITING..
120.             ..\"";
121.             $ns_ halt"
122.
123.             puts $tracefd "M 0.0 nn $val(nn) x $opt(x) y
124.             $opt(y) rp $val(rp)"
125.             puts $tracefd "M 0.0 sc $val(sc) cp $val(cp)
126.             seed $val(seed)"
127.             puts $tracefd "M 0.0 prop $val(prop) ant $va
128.             l(ant)"
129.
130.             puts "Starting Simulation...."
131.             $ns_ run

```

Kode Sumber 7.5 file .tcl untuk simulasi AODV 802.11a

```

1. BEGIN{
2.     recvs = 0;
3.     besaran_paket = 0;
4. }
5. {
6.     if (($1 == "s" || $1 == "r") && $4 == "RTR"){
7.         recvs++;
8.     }
9.     if (($1 == "s" || $1 == "r") && $4 == "RTR")
10.    {
11.        besaran_paket = besaran_paket + $8;
12.    }
13. }
14. END{
15.     printf("Routing Overhead : %.3f\n", besaran_paket/recvs);
16. }

```

Kode Sumber 7.6 Implementasi perhitungan RO

```

1. BEGIN {
2.     sendLine = 0;
3.     recvLine = 0;
4.     forwardLine = 0;
5. }
6.
7. $0~/^s.*AGT/{
8.     sendLine++
9. }
10.
11. $0~/^r.*AGT/{
12.     recvLine++
13. }
14.
15. $0~/^f.*RTR/{
16.     forwardLine++
17. }
18. END{

```

```

19.     printf"Ratio: %.4f\n", (recvLine/sendLine)*100;
20. }

```

Kode Sumber 7.7 implementasi perhitungan PDR

```

1. BEGIN {
2.   seqno = -1;
3.   # droppedPackets = 0;
4.   # receivedPackets = 0;
5.   count = 0;
6. }
7. {
8.   if($4 == "AGT" && $1 == "s" && seqno < $6) {
9.     seqno = $6;
10.  }
11.  if($4 == "AGT" && $1 == "s") {
12.    start_time[$6] = $2;
13.  } else if(($7 == "cbr" && ($1 == "r"))) {
14.    end_time[$6] = $2;
15.  } else if($1 == "D" && $7 == "cbr") {
16.    end_time[$6] = -1;
17.  }
18. }
19. END {
20.   for(i=0; i<=seqno; i++) {
21.     if(end_time[i] > 0) {
22.       delay[i] = end_time[i] - start_time[i];
23.       count++;
24.     }
25.     else
26.     {
27.       delay[i] = -1;
28.     }
29.   }
30.   for(i=0; i<=seqno; i++) {
31.     if(delay[i] > 0) {
32.       n_to_n_delay = n_to_n_delay + delay[i];
33.     }
34.   }
35.   n_to_n_delay = n_to_n_delay/count;
36.   printf("end to end delay : ")

```

```
37. print n_to_n_delay * 1000 ;
38. }
```

Kode Sumber 7.8 implementasi perhitungan E2E

Instalasi NS-2

Instalasi NS-2 dilakukan pada sistem operasi Ubuntu 14.04. Yang diperlukan untuk menggunakan NS-2 adalah melakukan instalasi dependensi dan *source code* ns-2.35. Sebelum melakukan instalasi NS-2 diperlukan beberapa dependensi agar NS-2 dapat dijalankan. Cara instalasi dependensi tersebut ditunjukkan pada perintah di bawah

```
1. sudo apt-get install build-essential autoconf automake
```

Kode Sumber 7.9 instalasi NS-2

Setelah semua dependensi terpasang selanjutnya adalah mengunduh *source code* ns-2.35 dengan cara yang ditunjukkan pada perintah di bawah

```
1. wget http://jaist.dl.sourceforge.net/project/nsnam/allinone/nsallinone-2.35/ns-allinone-2.35.tar.gz
```

Kode Sumber 7.10 mengunduh kode sumber ns-2.35

Lalu mengekstrak file dengan kode dibawah ini

```
1. tar -xvf ns-allinone-2.35.tar.gz
```

Kode Sumber 7.11 Ekstrak file NS-2

BIODATA PENULIS



Rezky Budi Prasetyo, lahir pada tanggal 5 Mei 1995 di Jakarta. Saat ini sedang menempuh perguruan tinggi di Institut Teknologi Sepuluh Nopember Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi pada tahun 2013. Terlibat aktif pada organisasi mahasiswa tingkat jurusan antara lain Departemen Pengembangan Sumber Daya Manusia(PSDM) Himpunan Mahasiswa Teknik Computer-Informatika (HMTIC) ITS sebagai staff tahun 2014 – 2015, Kadiv PSDM HMTIC tahun 2015 – 2016, sebagai staff Perkaptrans pada Schematics 2014, dan staff NST Schematics 2015, staff kaderisasi Keluarga Muslim Informatika(KMI) 2014 – 2015, staff ahli kaderisasi KMI 2015 – 2016 .

Apabila ingin bertanya lebih lanjut, silahkan kirim *email* ke rezky.kiki@gmail.com.